DTIC FILE COPY

2

NPS52-89-38

NAVAL POSTGRADUATE SCHOOL

Monterey, California

AD-A212 149

THESIS

VISUALIZATION
OF
HIGH-RESOLUTION DIGITAL TERRAIN

by

William Olin Breden
and
James Jeffrey Zanoli

June 1989

Thesis Advisor: Michael J. Zyda

Approved for public release; distribution is unlimited

DTIC
ELECTE
SEP 08 1989
S B D

Prepared for:
Commander
US Army Combat Developments Experimentation Center
Attention: ATEC-D
Fort Ord, California 93941

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NPS52-89-38	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (If applicable) 37	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION U.S. Army Combat Developments Experimentation Center 8b. OFFICE SYMBOL (If applicable) ATEC-D	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER ATEC 44-87
8c. ADDRESS (City, State, and ZIP Code) Fort Ord, CA 93941	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. | PROJECT NO. | TASK NO. | WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
VISUALIZATION OF HIGH-RESOLUTION DIGITAL TERRAIN

12. PERSONAL AUTHOR(S)
Breden, William O. and Zanoli, James J.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Master's Thesis	FROM ______ TO ______	June 1989	74

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	High-resolution digital terrain, 2D and 3D visual simulation, graphics workstation performance measurements, aerial photo displays, processed stereo imagery data.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)
We explore two methods for real-time generation of more realistic two and three-dimensional terrain displays than what are currently available on relatively inexpensive graphics workstations. The first method involves using a high-resolution terrain elevation database. The second method involves coloring and shading the terrain with gray-scale data obtained from associated aerial photography. Both methods were implemented with a three-dimensional simulator utilizing a high-resolution digital terrain database that was generated from processed F-14 stereo imagery. We describe our simulator, the High-Resolution Digital Terrain Model (HRDTM), listing its capabilities and graphics features. We also present how the system performs on a high-performance graphics workstation.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Michael J. Zyda	22b. TELEPHONE (Include Area Code) (408) 646-2305 | 22c. OFFICE SYMBOL 52ZK

Approved for public release; distribution is unlimited.

VISUALIZATION OF
HIGH-RESOLUTION DIGITAL TERRAIN

by

William Olin Breden
Captain, United States Marine Corps
B.S., Purdue University, 1978

and

James Jeffrey Zanoli
Captain, United States Army
B.S., United States Military Academy, 1982

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1989

Authors: William Olin Breden

William Olin Breden

James J. Zanoli

James Jeffrey Zanoli

Approved By: Michael J. Zyda

Michael J. Zyda, Thesis Advisor

Sehung Kwak

Sehung Kwak, Second Reader

Robert B. McGhee

Robert B. McGhee, Chairman,
Department of Computer Science

K.T. Marshall

Kneale T. Marshall,
Dean of Information and Policy Sciences

ABSTRACT

We explore two methods for real-time generation of more realistic two and three-dimensional terrain displays than what are currently available on relatively inexpensive graphics workstations. The first method involves using a high-resolution terrain elevation database. The second method involves coloring and shading the terrain with gray-scale data obtained from associated aerial photography. Both methods were implemented with a three-dimensional simulator. ~~utilizing~~ a high-resolution digital terrain database that was generated from processed F-14 stereo imagery. We describe our simulator, the High-Resolution Digital Terrain Model (HRDTM), listing its capabilities and graphics features. We also present how the system performs on a high-performance graphics workstation.

The High-Resolution Digital Terrain Model simulator was developed as part of a joint master of science degree thesis for Captain William O. Breden, USMC and Captain James J. Zanoli, USA. Captain Breden was responsible for database manipulation; the user interface; the two and three-dimensional terrain display, the magnification capability; the variable gamma ramp capability; ~~and~~ the camera (laser printing) capability. Captain Zanoli was responsible for file input/output; elevation contour map and aerial photo display, initial three-dimensional terrain display, and data panel performance measurements.

QUALITY INSPECTED 4

Accession For
NTIS GRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification

By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

Dr. Michael A. True, computer research manager at GeoSpectra Corporation, is to thank for providing the information on stereo imagery processing techniques and programs used by GeoSpectra Corporation in Ann Arbor, Michigan.

Captain Charles E. Phillips, USA and Lieutenant Gordon K. Weeks, USCG provided the routines that are used to draw the three-dimensional terrain in their submarine simulator. They also provided functions used to perform the timing calculations. Lieutenant Weeks' many hours of assistance in the modification and integration of his code with our simulator were appreciated more than he will ever realize.

Captain Mark A. Fichten, USA and Mr. David H. Jennings provided not only an excellent overview of the Moving Platform Simulator and its source code but also valuable advice and programming assistance d ring the early development of our simulator.

Our thanks also to Mr. Mike Tedeschi and Dr. Ignace Liu of the Combat Developments Experimentation Center (CDEC), Fort Ord, California. They provided the high-resolution terrain database and the direction for our research. We also should not overlook their most gracious funding of our research.

The enthusiastic support of our thesis advisor, Dr. Michael J. Zyda, was appreciated most of all. Our thanks to Dr. Zyda for providing us the graphics theory and programming background necessary to undertake such a project.

I. SIMULATOR DEVELOPMENT AND PERFORMANCE HISTORY

A. INTRODUCTION

The development of high-performance graphics workstations and digital terrain elevation databases has led to the production of some very good, and yet relatively inexpensive, flight and moving platform simulators [Ref. 1, 2]. The problem with these simulators, however, is the tradeoff that exists between realism and performance; essentially, the more realistic the simulation, the slower the simulator. In order to increase speed or performance, realism must be slighted.

Using the United States Naval Postgraduate School's Moving Platform Simulator (MPS) [Ref. 2] as our paradigm, we focus on current simulator capabilities and performance. We then seek to improve the realism of the graphical terrain display without significantly degrading system performance. In order to better understand the Moving Platform Simulator and the improvements we seek, we first review the system and how it has evolved.

B. FOGM MISSILE SIMULATOR

The Fiber-Optically Guided Missile (FOGM) simulator [Ref. 3] was the first in a series of simulators developed at the Naval Postgraduate School that led to the design of MPS. The FOGM was implemented in June 1987 on a Silicon Graphics, Inc. IRIS 3120 graphics workstation. The simulator presented a three-dimensional view from the missile as it flew over a 10 kilometer x 10 kilometer area of Fort Hunter-Liggett, California. In addition to the terrain, the simulator is also capable of displaying vehicles that are assigned initial headings and speed. Since the IRIS 3120 does not have the hardware to support real-time, double-buffered, hidden surface elimination, all drawing was accomplished using a scanline Painter's algorithm. All polygons are sorted from farthest away to closest to the viewer's position and then

drawn in that order to ensure that objects closer to the viewer are not obscured by distant objects. Vehicles are drawn after the terrain is displayed [Ref. 2:p. 4].

C. VEH VEHICLE SIMULATOR

The vehicle (VEH) simulator [Ref. 4], also implemented on a Silicon Graphics, Inc. IRIS 3120 graphics workstation, is an extension of the FOGM simulator. The VEH simulator, completed in December 1987, contains the same features for drawing terrain and vehicles as the FOGM with the exception that only terrain in the field-of-view is drawn using the scanline Painter's algorithm. Additionally, the VEH simulator allows for real-time selection and control of ground vehicles [Ref. 2: p. 5].

D. FOGM/VEH NETWORKING SIMULATOR

The FOGM/VEH NET allows networking between the FOGM and VEH simulators. This networking is accomplished over the Ethernet local area network that ties the graphics workstations together. The networking permits simultaneous vehicle position updating on the FOGM workstation while the user of the VEH operates a vehicle from his workstation. Networking also allows the missile flown on one workstation to be seen on the second workstation [Ref. 2:p. 5].

E. VEH II VEHICLE SIMULATOR

The VEH II simulator, completed in June 1988, was the result of not only software enhancements to the VEH but also porting the VEH from the IRIS 3120 to an IRIS 4D/70G and an IRIS 4D/70GT. VEH II has all the capabilities of the VEH simulator, but modifications allow the simulator to run on the newer hardware and under the MEX [Ref. 5] and 4Sight [Ref. 6] window management systems. Additionally, VEH II provides popup menus for all user selected options, the ability to add vehicles to the simulator at any time, and an option to save convoy status to a file that can be entered into the simulator with the appropriate popup menu selection [Ref. 2:p. 5].

F. MOVING PLATFORM SIMULATOR

The Moving Platform Simulator (MPS), completed in December 1988, is a combination of the FOGM and VEH II simulators. MPS was designed on and takes advantage of many features built into the hardware of an IRIS 4D/70GT graphics workstation. MPS allows the user to select a 10 kilometer x 10 kilometer grid area from a 35 kilometer x 35 kilometer database. The terrain color scheme is variable, and an efficient terrain drawing algorithm is able to display more terrain than earlier models by including distance attenuation. Z-buffering is used for hidden surface elimination. A selectable month and hour option determines the sun's location and sets the parameters for realistically lighted vehicles and terrain. The system also provides the FOGM missile the ability to track, target, and destroy vehicles. A collision detection scheme ensures that wrecked vehicles are rendered inoperative. Broadcast networking permits multiple simulations to run on different IRIS 4D/70GT graphics workstations [Ref. 2:p. 8].

G. TERRAIN DATABASE

The terrain database that all the simulators use was provided to the Naval Postgraduate School by the United States Army Combat Developments Experimentation Center (CDEC) at Fort Ord, California. The database is a special Defense Mapping Agency (DMA) database that consists of elevation and vegetation data in 12.5 meter increments in the 36 kilometer x 35 kilometer area encompassing Fort Hunter-Liggett, California. Each data point contains 16 bits. The three most significant bits are a vegetation code, which is ignored by the simulators, and the remaining 13 bits represent the elevation of the point measured in feet. The Moving Platform Simulator uses a 35 kilometer x 35 kilometer area with a resolution of 100 meters. This subset of the original database consists of 245,000 bytes; 100 samples per square kilometer, 1225 (35 x 35) square kilometers, and two bytes per sample

yield 245,000 (100 x 1225 x 2) bytes. At the time of this writing, MPS has just been modified to display the 12.5 meter resolution data [Ref. 2:pp. 9-10].

H. PERFORMANCE HISTORY

1. FOGM, VEH, VEH II Measured Performance

As the VEH II evolved from the FOGM, performance improved dramatically. The increased performance was a direct result of improved hardware. Performance measurements of the three simulators are based upon the number of frames drawn per second. As the simulators were ported from the IRIS 3120 to the IRIS 4D/70G, performance doubled. Performance doubled again after porting the simulators to the IRIS 4D/70GT. On the relatively slow IRIS 3120 workstation, the simulators would run at six to eight frames per second. Performance was measured at seven to 14 frames per second on the IRIS 4D/70G and 16 to 30 frames per second on the IRIS 4D/70GT. Typical measurements obtained from VEH and VEH II simulations running on an IRIS 3120, an IRIS 4D/70G, and an IRIS 4D/70GT are shown in Tables 1.1, 1.2, and 1.3 [Ref. 2:pp. 6-7].

2. MPS Measured Performance

Since MPS is much more complex than any of its predecessors, it is very difficult to compare it to them. However, a few measurements do offer excellent performance measurement benchmarks that can be used to compare MPS to other systems, past and present. The number of polygons drawn per frame and the number of frames drawn per second are two such measurements. Typical measurements obtained from MPS running on an IRIS 4D/70GT are shown in Table 1.4 [Ref. 2:pp. 61-63].

3. Simulator Realism

The goal of the work at the Naval Postgraduate School's Graphics and Video Laboratory is to develop accurate real-time three-dimensional graphics simulations

TABLE 1.1 ONE VEHICLE ON TERRAIN (FRAMES / SECOND)

SIMULATOR / MACHINE	15 DEGREE VIEW	55 DEGREE VIEW
VEH / IRIS 3120	8.0	6.5
VEH II / IRIS 4D/70G	14.0	7.0
VEH II / IRIS 4D/70GT	30.0	16.0

TABLE 1.2 NINE VEHICLES IN VIEW ON TERRAIN
(FRAMES / SECOND)

SIMULATOR / MACHINE	15 DEGREE VIEW	55 DEGREE VIEW
VEH / IRIS 3120	4.0	3.5
VEH II / IRIS 4D/70G	5.0	3.0
VEH II / IRIS 4D/70GT	10.0	6.0

TABLE 1.3 NINE VEHICLES ON TERRAIN, NONE IN VIEW
(FRAMES / SECOND)

SIMULATOR / MACHINE	15 DEGREE VIEW	55 DEGREE VIEW
VEH / IRIS 3120	6.0	5.0
VEH II / IRIS 4D/70G	12.0	7.0
VEH II / IRIS 4D/70GT	25.0	16.0

TABLE 1.4 MPS PERFORMANCE MEASUREMENTS ON IRIS 4D/70GT

DISPLAYING DETAILED TERRAIN

PLATFORM	ZOOM ANGLE	POLYGONS PER FRAME	FRAMES PER SECOND
ONE VEHICLE	55	763	8
ONE VEHICLE	15	403	14
NINE VEHICLES	55	1086	6
NINE VEHICLES	15	722	8
MISSILE 1500m	90	19801	< 1
MISSILE 1500m	10	3387	2

DISPLAYING ATTENUATED TERRAIN

PLATFORM	ZOOM ANGLE	POLYGONS PER FRAME	FRAMES PER SECOND
ONE VEHICLE	55	607	9
ONE VEHICLE	15	393	15
NINE VEHICLES	55	940	7
NINE VEHICLES	15	680	9
MISSILE 1500m	90	4152	2
MISSILE 1500m	10	816	7

on commercially available graphics hardware. The hardware must be powerful enough to handle the demands of a real-time system, but it must be relatively inexpensive. The $100,000 price range is considered inexpensive for our purposes. The graphics displays should provide as much detail as possible while still allowing a two to three frame per second update. The simulators developed to date all meet the two to three frame per second update requirement; however, detail is lacking in all the graphics displays. Figures 1.1 and 1.2 depict examples of current displays that are achieveable on the Moving Platform Simulator using a 100 meter resolution terrain elevation database. Recent improvements to MPS permit 12.5 meter resolution displays that offer quite a bit more realism than the 100 meter resolution displays. Figures 1.3 and 1.4 depict these recently obtained results. These simulators, however, fail to display cultural features and vegetation. It is this lack of information that inspired us to use a higher resolution terrain elevation database colored and shaded with its corresponding aerial photo gray-scale in our quest for a realistic real-time simulator. Our ultimate objective is to obtain image quality and detail similar to that of the black and white photos of Fort Hunter-Liggett depicted in Figures 1.5 and 1.6 [Ref. 1:p. 2].

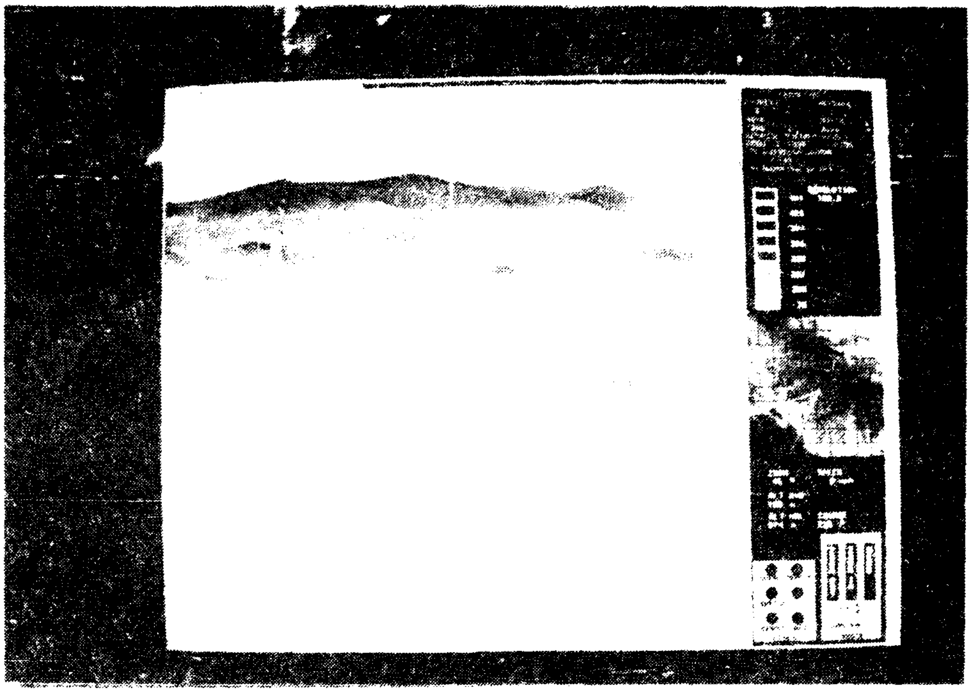


Figure 1.1 Moving Platform Simulator 100 Meter Resolution Display

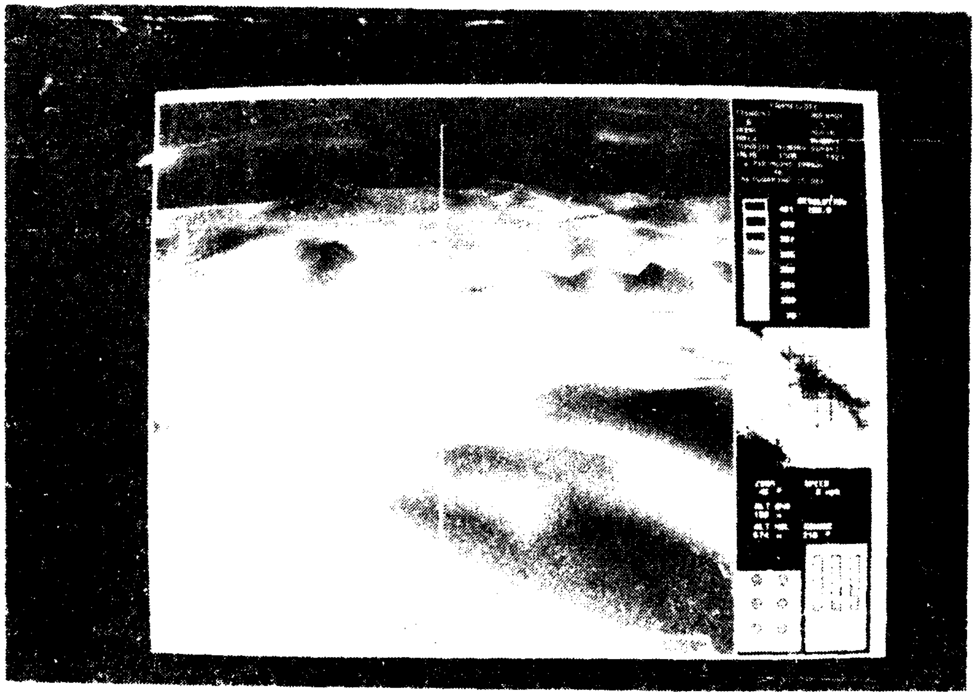


Figure 1.2 Moving Platform Simulator 100 Meter Resolution Display

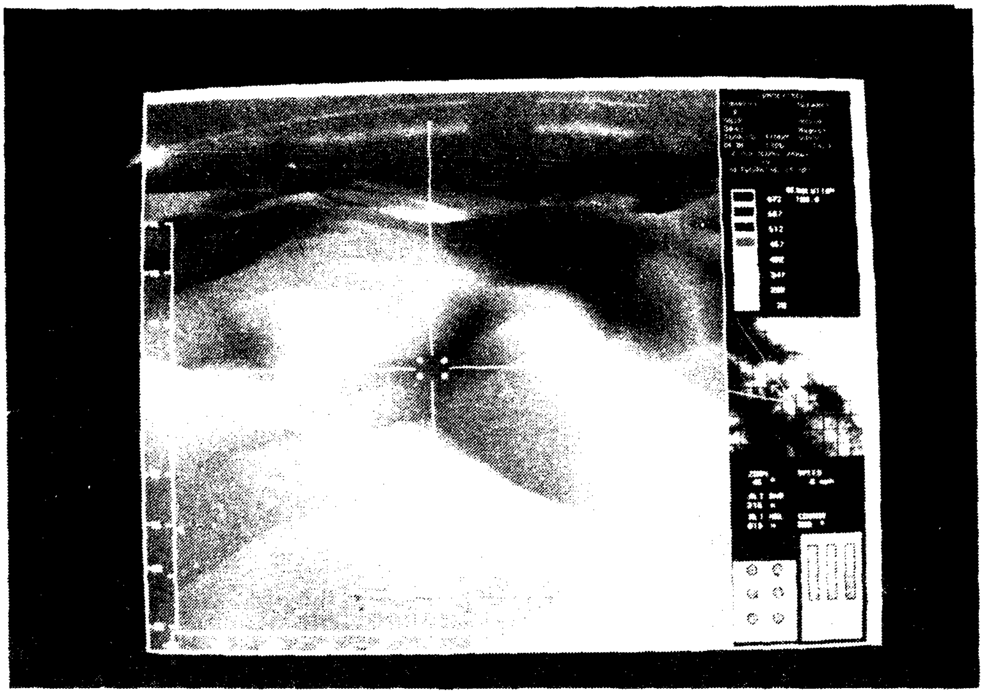


Figure 1.3   Moving Platform Simulator 12.5 Meter Resolution Display

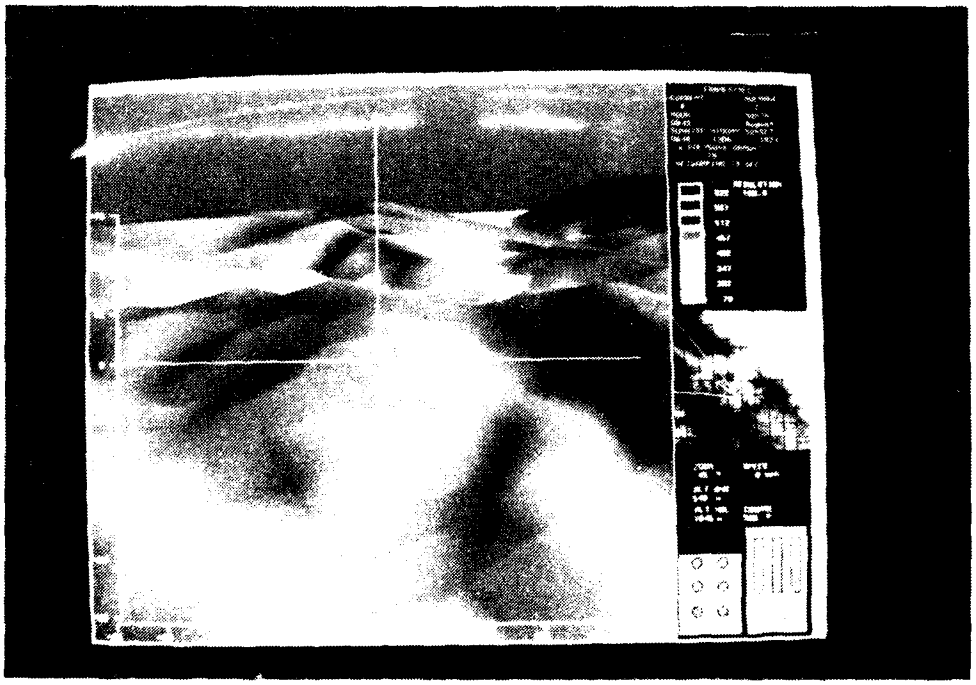


Figure 1.4   Moving Platform Simulator 12.5 Meter Resolution Display

Figure 1.5 Fort Hunter-Liggett Terrain Photograph

Figure 1.6 Fort Hunter-Liggett Terrain Photograph

II. HIGH-RESOLUTION DIGITAL TERRAIN MODEL DESCRIPTION

A. SYSTEM OVERVIEW

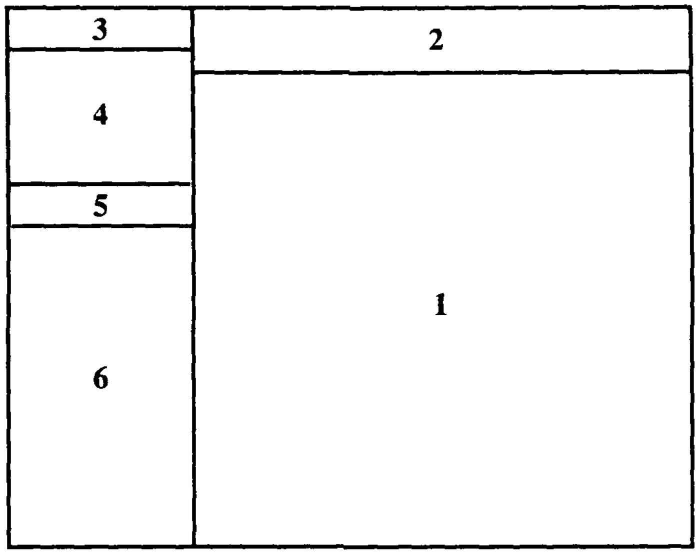
The High-Resolution Digital Terrain Model (HRDTM) simulator is a real-time moving platform simulator that models the nap-of-the-earth flight of a fiber-optically guided missile over three-dimensional terrain. The HRDTM simulator allows, to a limited extent, cultural feature and vegetation displays through textured terrain, terrain that is colored and shaded with its corresponding aerial photo reflectance (gray-scale) data. HRDTM research employs software methodology that seeks to improve the accuracy and realism of the images generated by the Moving Platform Simulator [Ref. 2] while maintaining real-time system performance. HRDTM was developed separately from the Moving Platform Simulator; but, it was intended for integration with MPS.

B. PROGRAMMING TOOLS AND ENVIRONMENT

HRDTM is designed and implemented on a Silicon Graphics, Inc. IRIS 4D/70GT graphics workstation. The workstation's programming environment is based on the AT&T Unix System V operating system and the 4Sight window management system. The environment includes an optimized C language compiler and a complete graphics library that provides routines for fast polygon fill, hidden surface elimination, and fast pixel access [Ref. 5, 6, 7].

C. SYSTEM FEATURES

The HRDTM simulator takes advantage of the fact that multiple window operations can run concurrently under the 4Sight Window Manager. HRDTM runs six windows concurrently. Figure 2.1 depicts the system's window layout. Three windows, however, provide nothing more than title information for the other three windows located beneath them. Of the three windows that display the bulk of the



1. Main Window
2. Main Title Window
3. Magnify Title Window
4. Magnify Window
5. Contour Map Title Window
6. Contour Map Window

Figure 2.1 System Window Layout

simulator's graphics, the main window is perhaps the most important. The main window allows for the static two-dimensional gray-scale aerial photo display or the dynamic three-dimensional missile view display of a selected area of terrain. The terrain selection is made from a second window, the contour map window, which displays either the gray-scale aerial photo or the elevation contour map of the entire terrain database. The final window, the magnify window, was designed primarily to display a magnified view of the terrain displayed under the cursor in the main window. Robustness in the code, however, permits the window to display the magnified version of any object located under the cursor positioned anywhere on the screen. Additionally, the window can be set to display the contour map's elevation legend, or it can be set to display a data panel that depicts the simulator's three-dimensional

drawing performance and missile's technical data such as speed, height, course, and view angle.

The HRDTM simulator is extremely simple to use. The entire simulation is driven by a popup menu that provides roll-off-the-side menu options. Besides the main menu options, the user can control missile parameters with the IRIS dial box while the simulator is in the three-dimensional terrain display mode. The data panel display permits the user to view a legend of the dial box and current missile parameters. A detailed description of system execution, to include a description of the user interface, is presented in Chapter 4.

III. OPERATING AREA

A. DATABASE PROCESSING

The high-resolution digital terrain database used in the HRDTM simulator was produced by GeoSpectra Corporation in Ann Arbor, Michigan. The database was obtained by processing F-14 stereo photography of Fort Hunter-Liggett, California. The F-14 carried the KS-87B framing camera that has a six inch focal length. Approximately 80 square kilometers of the Fort Hunter-Liggett area were photographed between 0900 and 1100 a.m. (Pacific Standard Time), 23 June 1987, from an altitude of 6,000 feet. A small sample of this photo coverage, approximately one square kilometer, was selected and processed into the HRDTM simulator's digital database using GeoSpectra's ATOM® software [Ref. 8].

ATOM® (Automatic TOpographic Mapper) [Ref. 9:p. 1] is a modularized photogrammetric program developed for a VAX mini-computer. It is designed to correlate eight bit digitized stereo photography and measure parallax for each image pixel. The digitized photography that the program analyzes is obtained by scanning stereo photos with an Optronics Scanner, and a Raster Technologies interactive display is used to locate and identify more than five image match points with known elevations. ATOM® consists of six functional modules [Ref. 10].

Module one, the control module, requires manual input of the camera's focal length and altitude, (or the photo scale), a minimum and maximum elevation, and more than five control points. This module presents an interactive display that permits the operator to roam through stereo pairs on a split screen. The control module performs the necessary computations to orient one photo of the stereo pair with respect to the other. At the end of this module, the operator is able to review the accuracy of the

stereo model's residual Y parallax and delta Z on point reference elevations. This module normally requires about an hour of the operator's time [Ref. 10].

Module two, the resample module, resamples the left and right images into epipolar space. This is a batch processing module that runs two to three hours on a VAX 11/750, given 81 megabyte images [Ref. 10].

Module three, the elevmap module, is a batch processing module designed to correlate pixels with eight bits of brightness. The speed of this module is variable and is a function of the parallax range [Ref. 10].

Module four, the editor module, allows the operator to interactively compare the stereo image data and elevation data. This module is used to interpolate correlation errors not caught by automatic editors [Ref. 10].

Module five, the ortho module, is a batch processing module that adjusts the X and Y location of each image pixel with respect to its elevation [Ref. 10].

Module six, the utilities module, includes routines that correct for lens and atmospheric distortions as well as the earth's curvature. This module was developed for processing small scale photos taken from the space shuttle [Ref. 10].

The output of ATOM® consists of two digital data files, an elevation file and a reflectance file. The first file consists of a very high density raster of terrain elevation data, and the second file consists of a co-registered raster of image brightness (gray-scale) data. The elevation file represents point elevations of each image pixel, either of bare ground, vegetation, or other features on the ground [Ref. 10].

B. FORT HUNTER-LIGGETT DATABASE

The terrain database that the High-Resolution Digital Terrain Model uses is a modified version of the GeoSpectra-produced database that was provided to the Naval Postgraduate School by the United States Army Combat Developments Experimentation Center (CDEC) at Fort Ord, California. The database consists of two files, an elevation data file and a corresponding aerial photo reflectance (gray-

scale) data file. The elevation file provides 0.1 meter elevation accuracy for data points that were sampled every 0.3 meters on the ground. The reflectance file provides the corresponding gray-scale data. Data points are sampled, in 0.3 meter increments, from the area formed by a square with the upper left corner at the geodesic coordinate N3972300, E667900 and the lower right corner at N3971000, E669200. These vertices were required in order to fit the entire sample data set into the coordinate system. Figure 3.1 depicts a graphical layout of the database while Figure 3.2 depicts the actual area of terrain on a 1:24,000 scale map produced from map sheets DMA 1755 I NW and AMS 1755 I SW - Series V895, dated 1949 and photoinspected in 1976. Data records are stored west to east, and successive records are stored north to south. The area is 1.3 kilometers wide and 1.3 kilometers high. Each elevation sample consists of 16 bits (two bytes) which when converted to decimal represents the elevation in tenths of meters. The decimal number 4953, for instance, would represent 495.3 meters. Each reflectance sample also consists of 16 bits (two bytes) and when converted to decimal represents the aerial photo's gray-scale level, an integer value between 0 (black) and 255 (white).

C. MODIFIED FORT HUNTER-LIGGETT DATABASE

The original database described above was modified to permit faster indexing into the array of terrain data and to eliminate unnecessay data which was consuming a huge amount of disk storage space. The first modification eliminated the zeros that were used to "pad" or "fill in" the square that surrounds the rectangular area of interest. The second modification reorganized the data; essentially, rotating the rectangular area of data that is shown in Figure 3.2 counter-clockwise until the resulting database shown in Figure 3.3 was achieved. Converting the two byte reflectance data file to a one byte data file constitutes the third modification. Inserting header information of six bytes representing the file type, row size, and column size of each file comprises the fourth and final modification. As a result of these

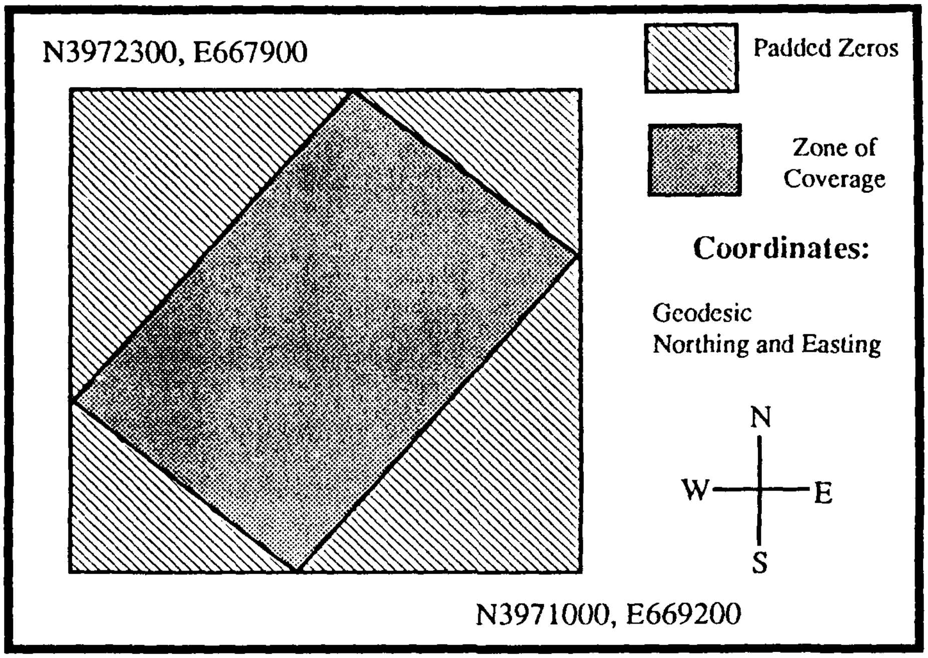


Figure 3.1 Original Database Layout

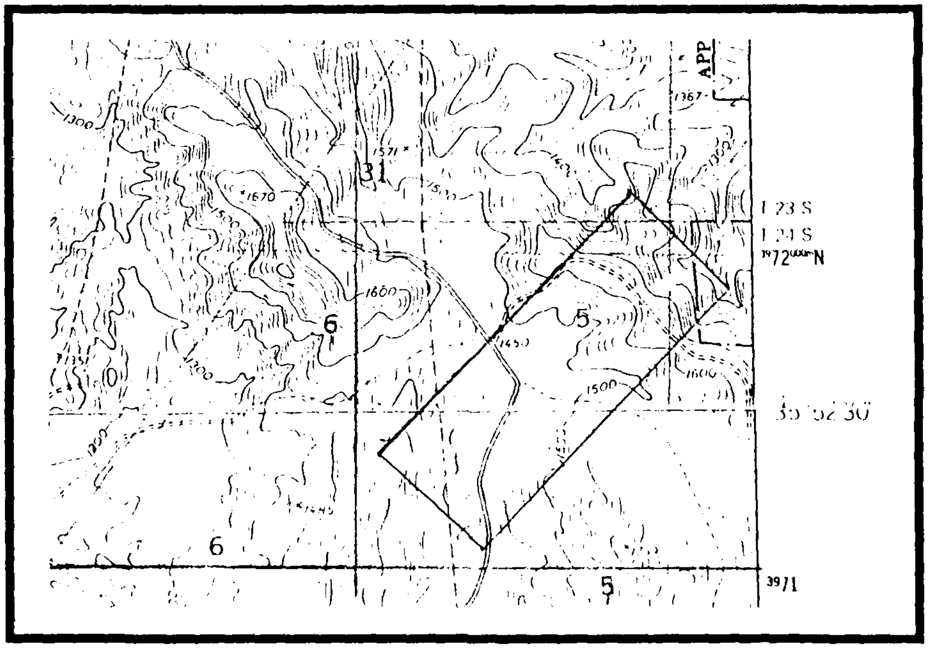


Figure 3.2 Area of Operation Overlay on 1:24,000 Map

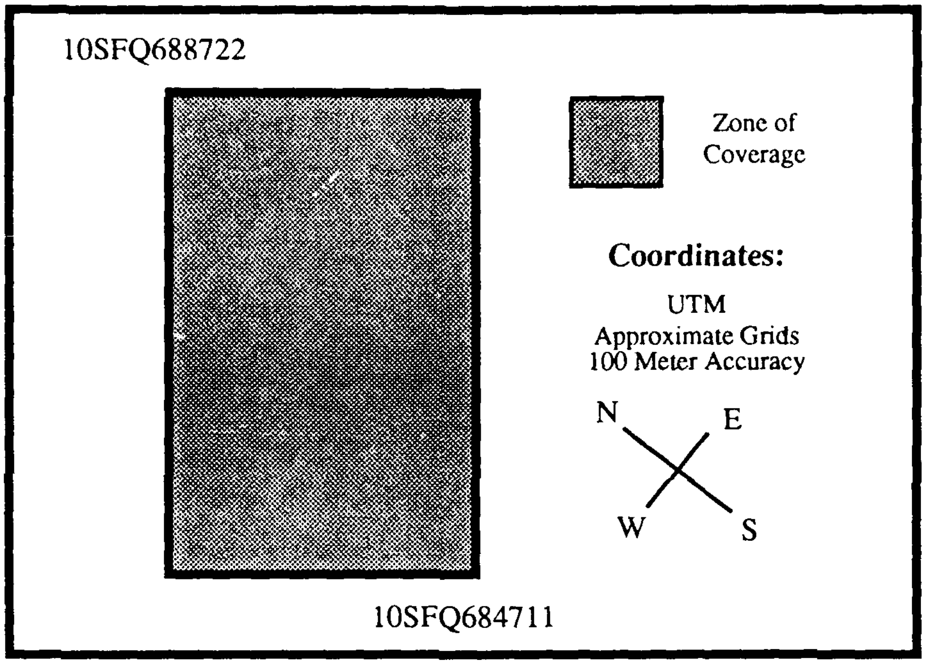


Figure 3.3 Modified Database Layout

modifications, the database is much more compact and easier to access. The modified elevation data file contains 12,819,210 (4001 rows of data x 1602 columns of data x 2 bytes per data sample + 6 bytes per header) bytes of data. The modified reflectance data file contains 6,409,608 (4001 rows of data x 1602 columns of data x 1 byte per data sample + 6 bytes per header) bytes of data. Thus, 19,228,818 bytes of disk space are required for storing the entire terrain database.

D. SELECTION METHODOLOGY

The HRDTM simulator is designed to allow the user to select any 960 sample x 960 sample (288 meter x 288 meter) area from the entire database. The user is not restricted to selecting an area that begins and ends on a one kilometer grid line as is the case in the Moving Platform Simulator. The 960 x 960 restriction is due to the main window size which is 960 pixels x 960 pixels and the manner in which the two-dimensional aerial photo is displayed using a system call that displays data samples using a fast pixel fill method.

E. COLOR SCHEME

The HRDTM simulator uses only one color scheme. All terrain, whether two-dimensional or three-dimensional, is colored and Gouraud shaded with its corresponding aerial photo reflectance data.

The elevation contour map can however be displayed using two different ramps, a gray ramp or a brown ramp. Both ramps consist of 16 shades of either gray or brown; the darker the shade, the higher the elevation. The 16 intervals are evenly spaced between the lowest and highest elevations in the database. The lowest and highest elevations are defined as constants in the program. They were obtained by running a separate search program on the elevation data file. This information can be added as part of the elevation data file header; thus, eliminating the need for constants and permitting the use of additional data files.

IV. GRAPHICS DISPLAY SPECIFICS

A. SUPPORTING GRAPHICS HARDWARE/SOFTWARE

1. IRIS Display Memory

The display memory on the IRIS-4D/70 GT is organized as a set of 96 bitplanes. A bitplane contains one bit of information for each pixel on the screen; therefore, up to 96 bits of information can be saved for each pixel on the screen. The screen is 1,280 bits wide and 1,024 bits high, which implies that each bitplane holds 1,310,720 (1,280 x 1,024) bits of information. These bits store color information as well as information about depth, overlays and underlays, and an alpha channel [Ref. 7: p. 4-1].

2. Color/Multimap Modes

The IRIS graphics workstation provides RGB and color map modes. RGB mode permits the programmer to dynamically create colors by setting the red, green, and blue color gun intensities of desired pixels [Ref. 7:p. 4-2]. Color map mode, on the other hand, requires the user to predefine colors that are later indexed from a table of 4096 possible entries [Ref. 7: p. 4-13]. Multimap mode divides the system's color map table of 4096 entries into 16 maps of 256 entries [Ref. 7: p. 4-21].

3. Double Buffering

Double buffering is a technique used for smoothing motion between images that change with time. This capability is achieved by hardware in the IRIS graphics workstation. The system's standard bitplanes are divided into two halves; one half is displayed (visible buffer), while drawing is performed in the other half (invisible buffer). The buffers are swapped when the drawing is complete, and the previously invisible buffer (the next frame) becomes visible, and the previously visible buffer becomes invisible and available for drawing the following frame [Ref. 7: p. 6-1].

4. Z-Buffering

The IRIS 4D/70GT graphics workstation contains special hardware that provides hidden surface elimination. This hardware consists of a bank of 24 bits (24 z-buffer bitplanes) that store depth (Z coordinate) information. When the IRIS is in z-buffer mode, the Z coordinate for each pixel is stored as a 24 bit value in the bitplanes. When a pixel is drawn, its new Z value, the distance from the object to the viewer's eye, is compared to the existing Z value. If the new Z value is less than or equal to the current Z value, the new color and Z values for that pixel are written into the bitplanes. Otherwise, the color and Z values remain unchanged. As a result, only parts of the image that are actually visible to the viewer are displayed on the screen. Note that the values in the z-buffer always represent the distances of the objects closest to the viewer [Ref. 7:p. 8-3].

5. Gamma Ramp

The IRIS graphics workstation provides a gamma correction capability, the ability to equalize monitors with different color characteristics or to modify the color warmth of the monitor. The gamma factor actually represents the nonlinearity of the monitor. Varying this factor essentially effects image contrast. The *gammaramp* function varies the gamma factor, effecting only the display of color, not the values that are written in the bitplanes. It also effects the entire screen and all running processes. It stays in effect until the system hardware is reset or another call to the *gammaramp* function is made [Ref. 7: p. 4-24].

6. Overlays

Information can overlay, be drawn over, the standard pixel contents of the current buffer. The IRIS achieves this capability through its overlay bitplanes that supply additional bits of information at each pixel. The significance of overlays is that overlay bitplanes can be displayed, modified, and then redisplayed without disturbing the current drawing [Ref. 7: p. 11-1].

7. Gouraud Shading

Gouraud shading is a means for varying the color across a polygon. The shading is achieved, first, by linearly interpolating the colors of each vertex along the edges connecting them. Then the interpolated colors along the edges are interpolated again across the interior of the polygon. Gouraud shading can be accomplished in RGB or Color Map mode. In RGB mode, the interpolation is linear in all three components, the red, green, and, blue intensities [Ref. 7: p. 4-7]. In color map mode, the color map index is interpolated [Ref. 7: p. 4-15].

8. Fast Pixel Access/Display

The IRIS graphics workstation supports high performance pixel access and display. The system function *rectread*, given the lower-left and upper-right corners of a rectangle, reads a rectangular array of pixels from a window and stores it in a given array [Ref. 7:p.10-3]. This array of pixels can be displayed with the system function *rectwrite* [Ref. 7:p. 10-4]. Note that the system function *readsource* does not work as intended; it fails to determine the source of pixels read by *rectread*. This is a software bug that should be corrected in version 3.2 of the IRIS operating system. Additionally, *rectread* does not perform according to specifications. It appears to read above and to the right of the point specified by the programmer [Ref. 7:p. 10-4].

B. MODELING TECHNIQUES

1. FOGM Parameters

The fiber-optically guided missile parameters include height, speed, tilt (look-down) angle, and course (heading). All parameters are user controlled and can be changed interactively through the IRIS dial box. Height represents the height of the missile, in meters, above the ground. Missile height ranges from one to 400 meters above ground. The speed of the missile is measured in kilometers per hour. Missile speed ranges between a negative and positive four kilometers per hour. A negative

speed allows the missile to move backwards, while a positive speed advances the missile. The missile has a built-in 45 degree field-of-view. This field-of-view can be adjusted to point anywhere between just below the horizon, one degree, and directly below the missile, 89 degrees. The course, or missile heading, is also adjustable. The missile can rotate 360 degrees. The measurements, made in degrees, are made relative to the missile's initial heading of zero degrees. The missile's location and field of view are continually updated in the contour map window based upon the missile's current speed and heading.

2. Timing

The HRDTM simulator continually updates the missile's location and drawing performance. In order to update both of these items, the simulator must know the time that has elapsed since its last update. This elapsed time is calculated in the *process_time_difference* function which queries the system function *times*. This system function returns the current number of clock cycles which is then subtracted from the value obtained during the previous program loop. This difference divided by the clock rate results in the elapsed time. Note that the simulator's time function is initialized in the program's *main* routine. Figure 4.1 depicts the elapsed time calculation. To calculate the distance covered by the missile during the elapsed time, the elapsed time is multiplied by the missile's speed; recall that *distance = rate x time*. We then calculate the X and Z components of total distance by multiplying distance by the sine and cosine of the missile's heading. The X and Z components are then added to the missile's current gridX and gridY components, thus updating the missile's location. Figure 4.2 shows the code that updates the missile's location based upon its current speed.

3. Terrain Drawing Algorithm

The terrain drawing algorithm is rather complicated since it only draws terrain that is in the missile's field-of-view. The algorithm permits only the terrain in the

```
/* This C code was provided by LT. Gordon K. Weeks, USCG. */

/* Global variables for time keeping */

long start_time;
struct tms timeinfo;

/* Process the loop time difference */
/* Called by updatepositions() */

/* Calculates the time expended during the last simulation loop and returns */
/* the elapsed seconds.                                                      */

float process_time_difference()

{
  struct tms timeinfo;  /* System time information    */
  float elapsedsec;     /* Returned time value         */
  long lastsec;         /* End time for simulation run */
  long elapsedhz;       /* Elapsed machine cycles      */

  /* start_time, global start time                     */

  lastsec = times(&timeinfo);
  elapsedhz = lastsec - start_time;
  elapsedsec = (float)(elapsedhz)/(float)HZ;
  start_time = lastsec;

  return(elapsedsec);
}
```

Figure 4.1 Process Time Calculation

```
/* This C code represents a small segment of the drawManeuver */
/* function that displays the three-dimensional perspective view */

/* Reset the variables used to store the 2 directional components of travel */

delta_x = 0;
delta_y = 0;

/* Distance = Rate x Time */
/* Convert km/hr to km/sec then scale the distance to coordinate system */
/* Elapsed_time is the number of seconds since last missile update */

distance_covered = speed / 3600.0 * 3333.33  * elapsed_time;

/* Compute change in drawing position */
/* Direction of travel is tied into the viewing direction */

if (speed != 0)
  {
    delta_x = distance_covered * cos (view * PI / 180);
    delta_y = distance_covered * sin (view * PI / 180);
  }

/* Update the missile's location by updating appropriate globals */

gridX = gridX + delta_x;
gridY = gridY + delta_y;

/* Display the terrain then loop through the entire process again */
```

Figure 4.2 Missile Location Update

missile's 45 degree horizontal and vertical fields-of-view to be displayed. This is done by drawing the three-dimensional scene in five degree arcs, referred to as sectors. As depicted in Figure 4.3, each sector subtends an arc of five degrees with the closed end of the arc located at the drawing point and the open end extending out to the farthest points visible from the viewing point. This method allows the increase/decrease in the number of sectors drawn based on the viewing tilt angle required to cover a drawing region slightly larger than the viewing region. Since the missile can achieve altitudes to 400 meters, the algorithm must also account for a look-out as well as a look-down capability. At the same time, distance attenuation must be considered. Figure 4.4 depicts the three-dimensional terrain drawing algorithm.

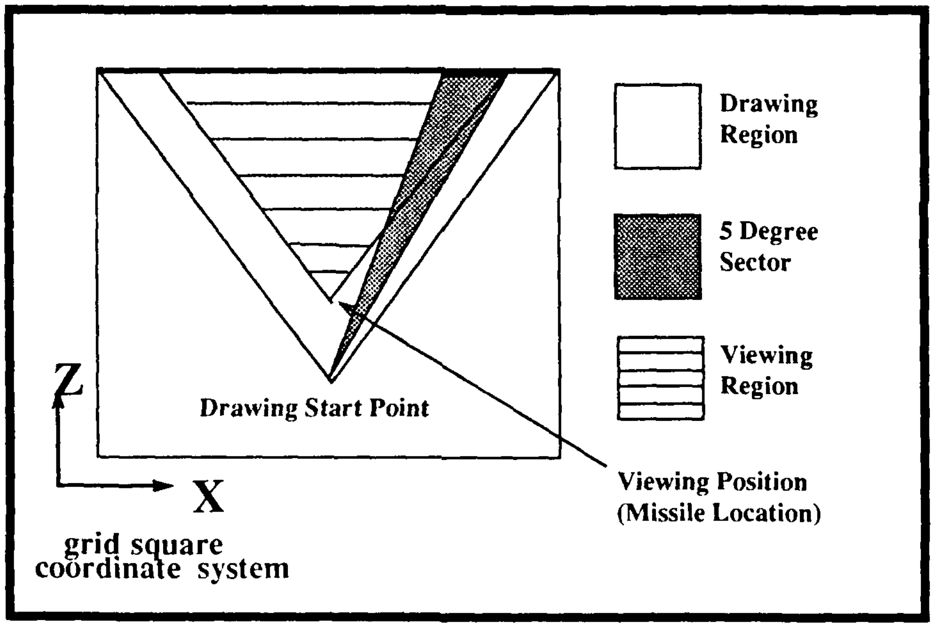


Figure 4.3 Sector Description

In HRDTM, terrain is drawn in double-buffer mode using the triangular mesh drawing routine provided by the graphics library. The mesh routine is used in the low level drawing routines to draw square terrain segments as two triangles. The terrain

Three-Dimensional Terrain Drawing Algorithm

1. Update the missile's location based upon its speed and the elapsed time since its last update.

2. Determine the elevation of the terrain immediately beneath the missile.

3. Calculate the maximum drawing distance, the furthest point from the missile that can be drawn.

4. Adjust the drawing distance and the number of sectors drawn in order to reduce the drawing time.

5. Ensure that the drawing distance does not exceed the maximum drawing distance determined in step 3.

6. Set the high-resolution and medium resolution drawing distances to create a distance attenuation effect.

7. Set the viewing perspective's viewing angle, aspect ratio, and clipping planes.

8. Set the lookat function's parameters to reflect current missile parameters and information determined above.

9. Compute an offset point to be used as a drawing start point. Offset is used to clear up clipping along side of 3D view.

10. Draw the 3D, perspective view in sectors.

11. Update data panel.

12. Return to step 1.

Figure 4.4 Three-Dimensional Terrain Drawing Algorithm

is Gouraud shaded by providing each triangular vertex its corresponding reflectance value and then linearly interpolating these values across the edges and faces of each triangle. The resulting view of the terrain is a three-dimensional, gray-scale, Gouraud shaded perspective view generated by the graphics library's *perspective* and *lookat* functions. The missiles's vantage point and viewing reference point are controlled or modified by varying the viewer's coordinates and viewed target's reference points within the *lookat* function.

The missile's horizontal and vertical view angles are fixed at 45 degrees. The distance from the missile to the ground plane, along a 45 degree angle, is calculated and used to create a pyramidal viewing volume. Figure 4.5 depicts the viewing volume with respect to the elevation data array which represents a partitioned ground plane. Only the terrain within this volume is drawn in order to minimize the number of polygons drawn and, thus, maximize the frames per second drawing update rate. By varying the size and position of this viewing volume, we simulate the effect of moving over three-dimensional terrain. Figure 4.6 depicts the effects, on the viewing volume, of varying the missile's course or heading. Figure 4.7 depicts the effects of varying the missile's height, and Figure 4.8 depicts the effects of varying the missile's tilt angle.

Additionally, we add to the realism of the view by simulating distance attenuation, the blurring or fading of distant terrain. Objects close to the viewer are drawn in detail, while objects located further away from the viewer are drawn in less detail. We achieve this effect by drawing close objects using every point in the database (high-resolution), semi-distant objects using every other point in the database (medium-resolution), and distant objects using only every fourth point in the database. At ground level, the terrain is drawn in high-resolution mode out to 75 meters. Medium-resolution is displayed between 75 and 200 meters, while low-resolution is displayed between 200 meters and the maximum distance drawn, the distance to the farthest point in a 960 x 960 array of the selected area's terrain data.

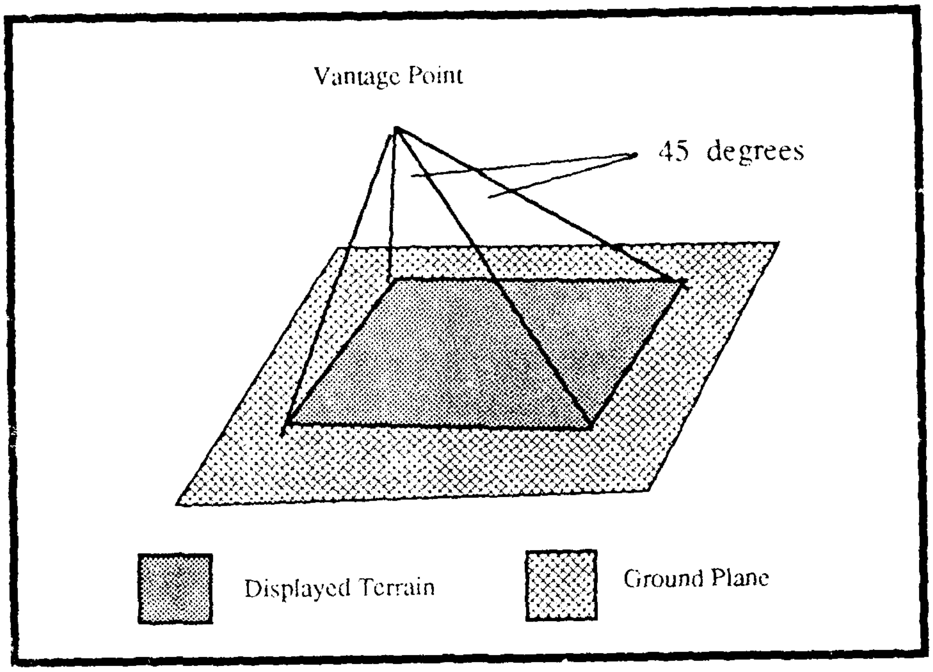


Figure 4.5 Viewing Volume Over Ground Plane

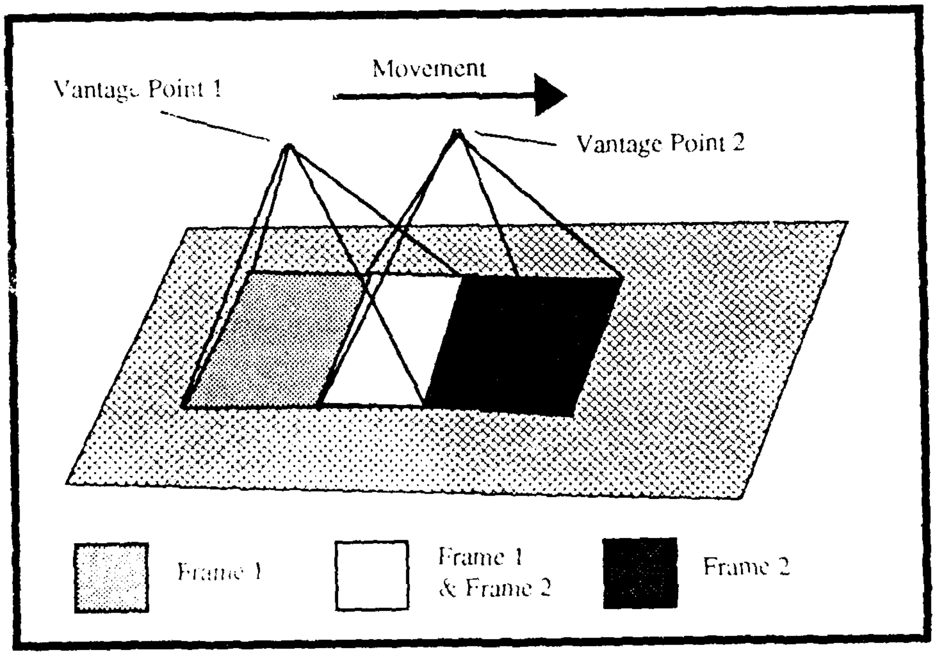


Figure 4.6 Simulated Movement Through Viewing Volume Displacement

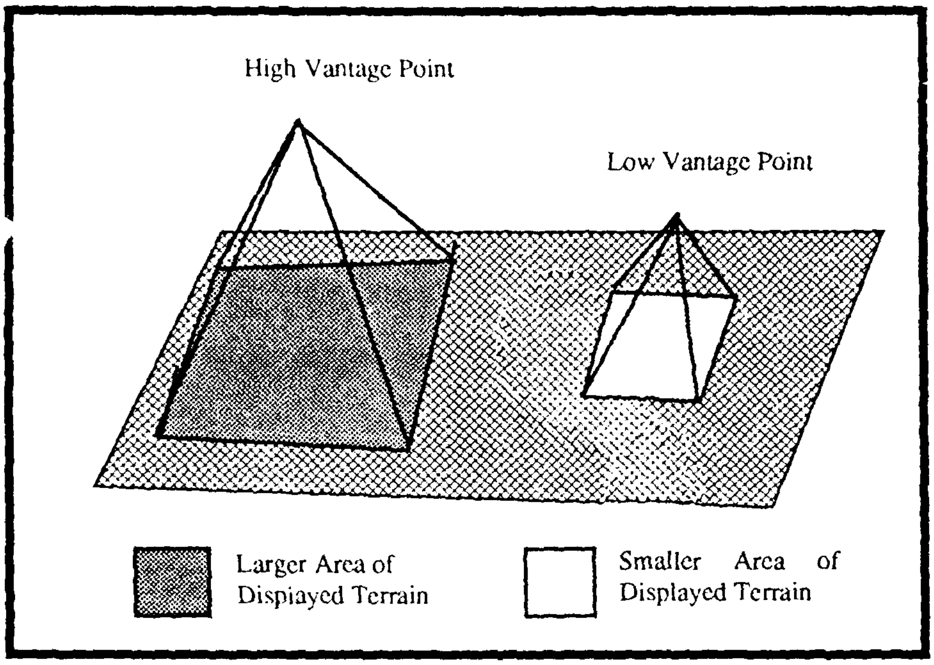


Figure 4.7 Missile Height Variation Effect

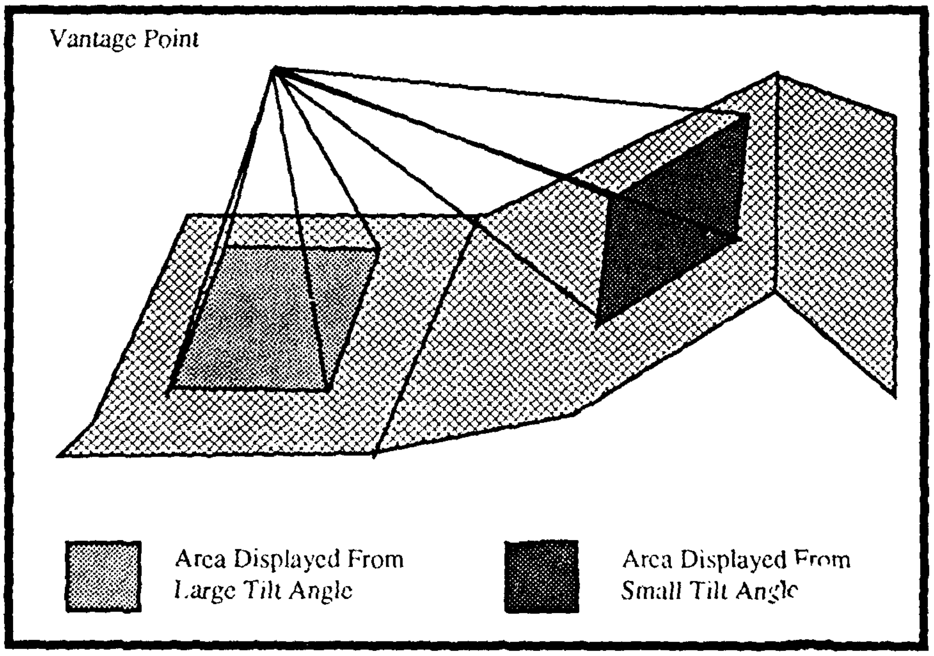


Figure 4.8 Missile Tilt Angle Variation Effect

As the viewing height and the tilt angle increase, the maximum drawing distance decreases causing less distant terrain to be displayed. Simultaneously, resolution boundaries are adjusted to account for these changes. Figure 4.9 depicts the distance attenuation drawing concept, while Figure 4.10 depicts the boundary adjustment algorithm. The boundary adjustment algorithm calls for the boundaries to decrease one meter for every three meter increase in elevation.

The maximum drawing distance adjustment is based upon a simple algorithm that is depicted in Figure 4.11. We simplify calculations by restricting the viewing look-down angle to a value between the critical angles of 22.5 degrees and 68 degrees. This implies that a 45 degree field-of-view permits a view between one and 89 degrees below the horizon. This restriction permits us to measure the distance between the missile, our vantage point, and the ground plane along a 45 degree azimuth. With this distance, we then solve for the ground distance (our maximum drawing distance) to our target view point.

Since the terrain is drawn as a series of overlapping five degree sectors, an algorithm also had to be developed to display a sufficient number of sectors to fill the screen as the viewing height and look-down angle increase. The viewing position first had to be adjusted for ground level because a 45 degree field-of-view drawing does not fill the screen entirely. Offsetting the viewing position forward of the drawing's starting point eliminates the problem. Figure 4.3 depicts this offset and the resulting view. As the tilt angle increases, the number of sectors drawn must also increase in order to fill the screen. A number of trial runs enabled us to determine the critical angles where the number of sectors drawn did not completely fill the screen. We use these angles to decide when we should increase the number of sectors that we draw. This method simplifies the drawing algorithm tremendously. Figure 4.12 depicts the algorithm that determines the required number of sectors to be drawn as the tilt angle increases.

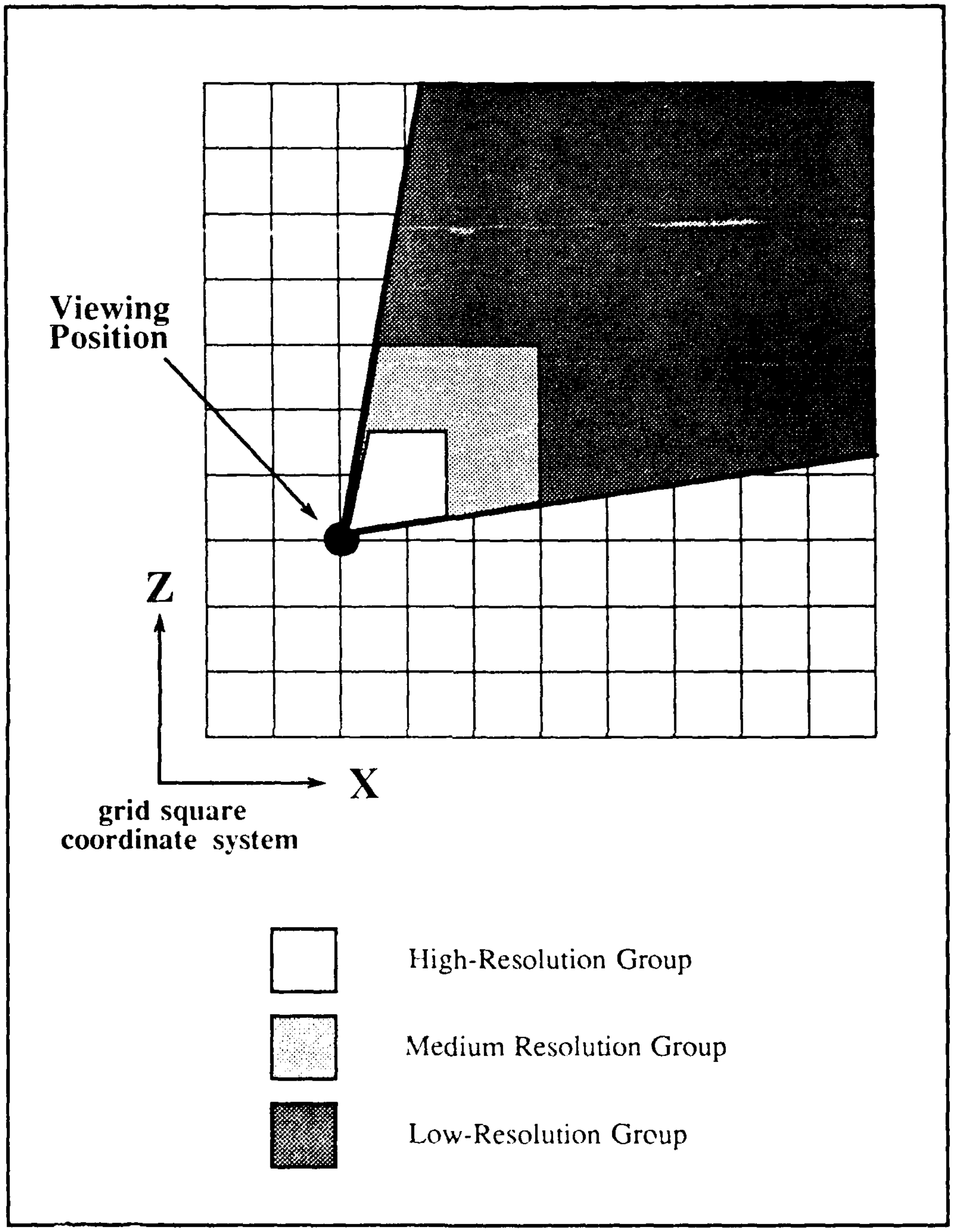


Figure 4.9 Distance Attenuation Drawing Concept

```
/* High and medium resolution drawing distance is based */
/* on the maximum viewing distance and viewing height. */
/* Hi_Res may be displayed to 75 meters, and Med_Res */
/* may be displayed to 200 meters. Distances greater than */
/* 200 meters are drawn in Low_Res. */


        Hi_Res = (75 - (viewer_height / 3)) / 2 * 2;
        if (Hi_Res < 0)  Hi_Res = 0;

        Med_Res = (200 - (viewer_height / 3)) / 2 * 2;
        if (Med_Res < 0)  Med_Res = 0;
```

Figure 4.10 Boundary Adjustment Algorithm

```
/* Determine max draw distance regardless of viewing  */
/* direction. Based upon 960 x 960 data array size. */

if (gridX < 481) tmp_x = 960 - gridX;
else  tmp_x = gridX;

if (gridY < 481)  tmp_y = 960 - gridY;
else tmp_y = gridY;

Max_Draw_Distance = sqt (tmp_x * tmp_x + tmp_y * tmp_y));

if (tilt_angle < 23)  Draw_Distance = Max_Draw_Distance;
else
{  Ground_Center_Distance = (((viewer_height) * SCALE) /
                              tan ((tilt_angle - 22.5) * PI / 180)));
   if (tilt_angle > 45)
   Draw_Distance = (Ground_Center_Distance / (SCALE * 2));
   else
   Draw_Distance = Ground_Center_Distance;
}
```

Figure 4.11 Drawing Distance Adjustment Algorithm

```
/* Terrain is drawn in a circular pattern consisting of 5 degree */
/* sectors. In order to speed the drawing process, only those */
/* sectors within the field of view are displayed. Critical */
/* angles were determined from trial and error. Critical angles */
/* occurred when terrain would not completely fill the screen */
/* because an insufficient number of sectors were drawn; the */
/* minimum number of sectors being 6, calculated at ground */
/* level. */


    if (tilt_angle < 23) NumberOfSectors = 6;
    else
      if (tilt_angle > 68) NumberOfSectors = 36;
      else NumberOfSectors = (short) (tilt_angle / 3.5);
```

Figure 4.12 Sector Calculation

C. SIMULATOR EXECUTION

1. System Start-up

The executable program file is **hrdtm**. While in the directory containing the executable file, type **hrdtm** to start program execution. Six windows open and run concurrently during the HRDTM simulator operation.

After the opening display, the contour map window is cleared and an elevation contour map of the entire database is displayed. Figure 4.13 depicts the elevation contour map in the lower-left corner of the screen. The double-buffered contour map window has the black and white aerial photo of the entire database drawn in the back buffer. Since the elevation contour map and the aerial photo only need to be drawn once, we found that double buffering enables us to draw each in a separate buffer; a menu option permits toggling between the two buffers with no drawing time cost. Figure 4.14 depicts the aerial photo of the entire database. Note that the photo and contour map are drawn using every third point in the database. Since the database is so much larger than the contour map display window, displaying every point in the

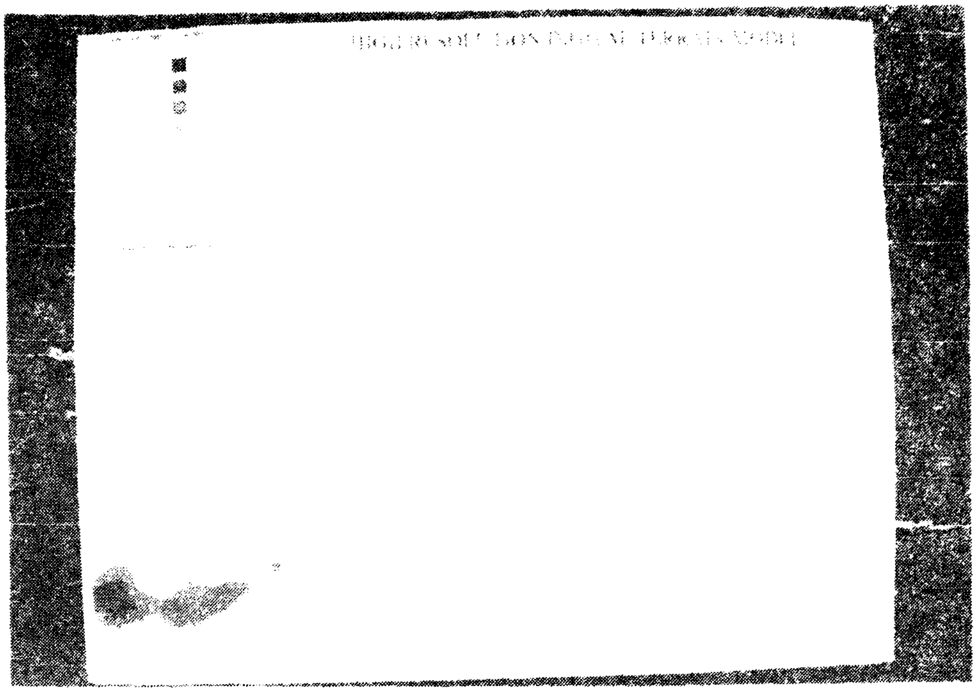


Figure 4.13 Elevation Contour Map Display

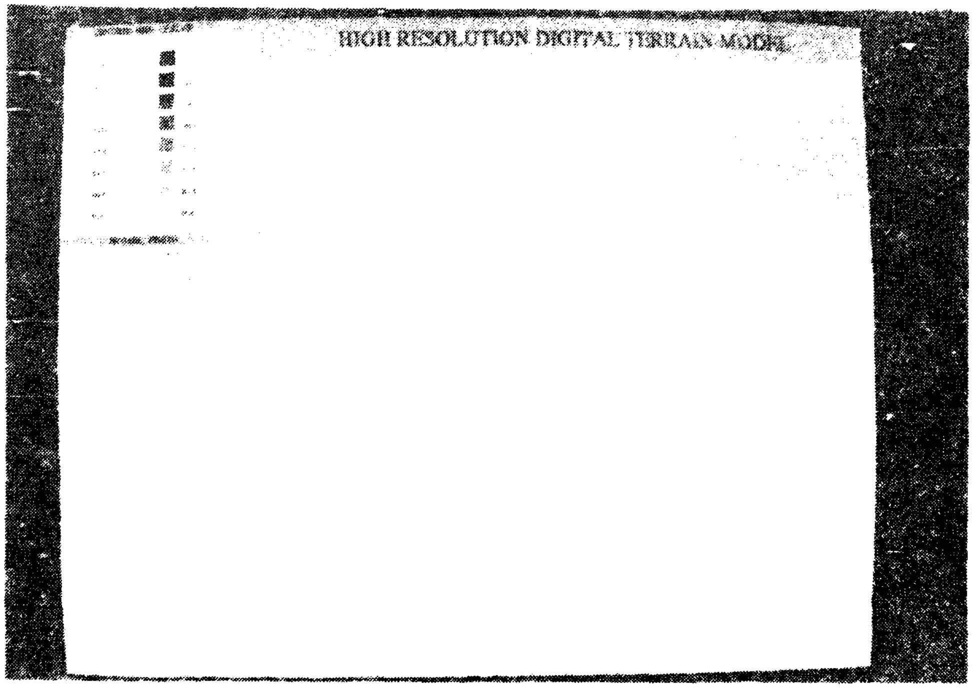


Figure 4.14 Aerial Photo and Map Legend Display

database would cause much of the data to overwrite or overdraw itself. Thus, displaying every data point only slows the drawing process; it does not provide any more visible information. Displaying anything less than every third data point, however, results in unfilled pixels and a visible loss of information.

Once the elevation contour map and aerial photo are drawn, the contour map's elevation legend is displayed in the second of the three primary windows. The legend is depicted with the aerial photo in Figure 4.14. This window is referred to as the magnify window in the source code. The legend depicts the colors and associated elevations, in meters, of the 16 contour intervals.

The user must then select an option from the system menu in order to continue program execution. The system menu is always invoked by depressing the right mouse button. The system menu, a popup menu with roll-off-the-side menu options, is the primary source of user input. The system menu is depicted in Figure 4.15. Note that the user can select any menu option at any time during program execution. However, if the user chooses a menu option that is not allowed or meaningful when the menu is displayed, nothing happens.

2. Terrain Selection

The simulator's main window, the map window, acts as a message output window as well as a means to display the two and three-dimensional views of selected areas of terrain. As depicted in Figure 4.14, the user is prompted to select the right mouse button for the main menu once the contour map and the legend are displayed. The user can then select an area of terrain by choosing the *select area* menu option. After this option is selected, a red box appears in the contour map window. Figure 4.16 depicts this selection box. The box can be placed over the desired area by moving the mouse. Selecting the left mouse button loads that area's elevation and reflectance data. Status messages printed to the main window keep the user appraised of the system's progress. Once the data is loaded, the user can

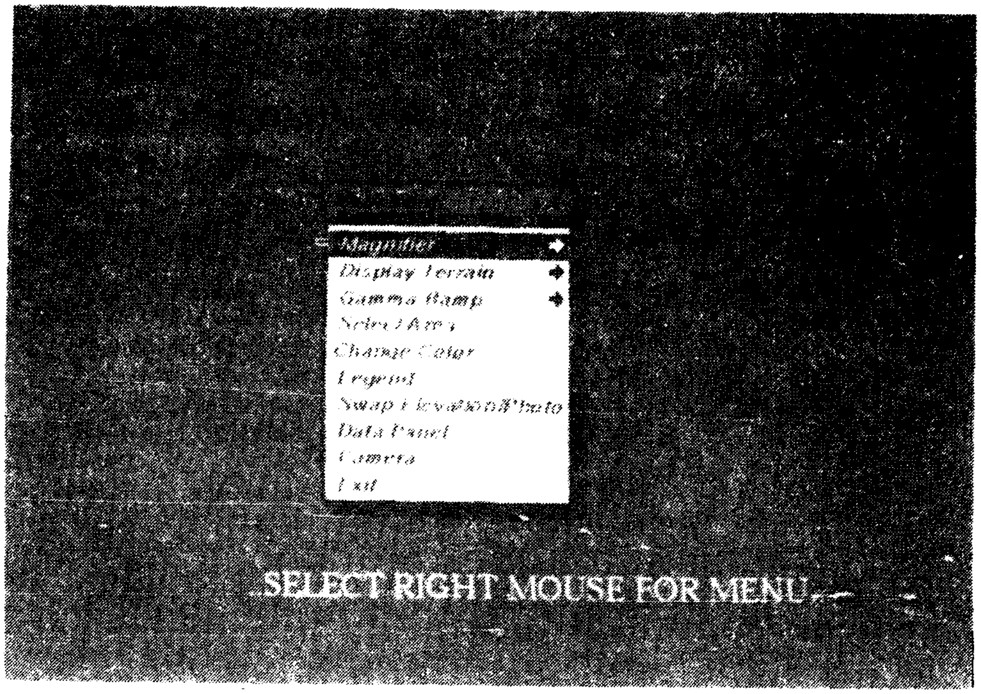


Figure 4.15   System Menu

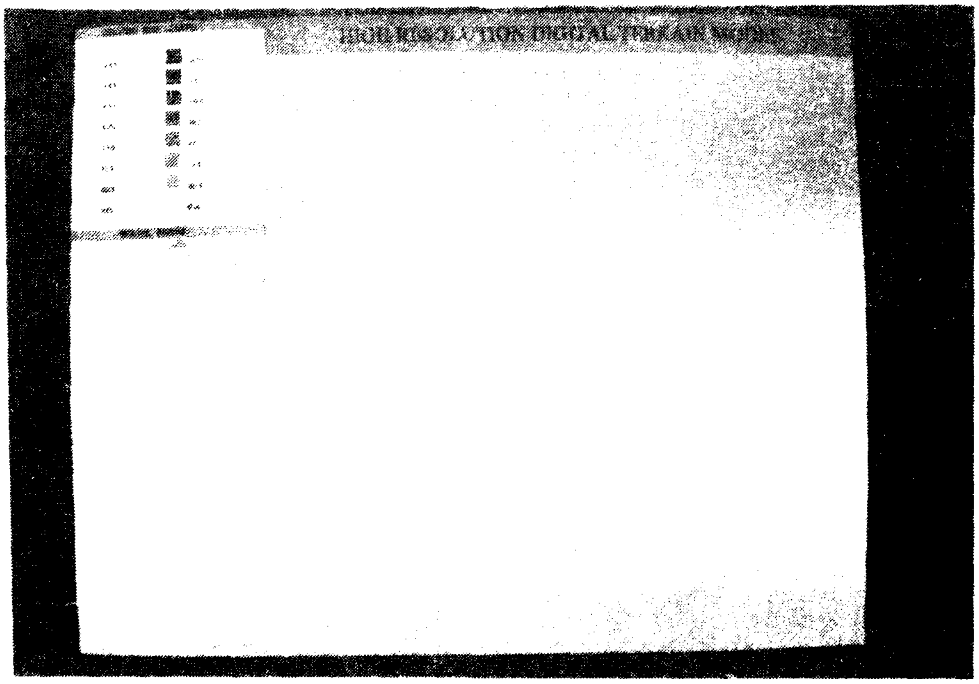


Figure 4.16   Operational Area Selection Box

display the area in two or three dimensions by making the appropriate menu selection. The *select area* option can be exercised any time during simulator execution.

3. Terrain Display

Once the area of terrain is selected, display the terrain by choosing either the *2D* or *3D* roll-off-the-side menu option to the main menu's *display terrain* option. The *2D* menu option displays, in the main window, a static two-dimensional aerial photo of the selected area of terrain. For the two-dimensional display of the selected area of terrain, such as the displays depicted in Figures 4.17 and 4.18, the simulator uses the system function *rectwrite* to quickly display the array of reflectance values which represent the gray-scale levels of the selected area of terrain. Note that rectwrite requires the reflectance data be stored in row-major order in the array to avoid creating two similar images reduced in size.

The *3D* menu option presents a nap-of-the-earth three-dimensional perspective view of the terrain from a fiber-optically guided missile. This view is changed, in real time, by varying the missile's parameters. Figures 4.19 and 4.20 depict sample three-dimensional perspective views.

4. Data Panel

Missile parameters can be observed in the data panel that is displayed in the magnify window by selecting the *data panel* menu option. Figure 4.21 depicts the data panel. The data panel not only shows missile parameters such as height above ground, course heading, speed, and tilt angle but also system performance data, specifically, the number of polygons drawn in order to create the three-dimensional view and the rate that these views are being updated (frames per second). Additionally, the data panel provides a legend for the dial box. The dial box, depicted in Figure 4.22, provides the user the means by which he can effect the missile's parameters. The data panel is updated with each frame; therefore, the panel provides current drawing data and missile parameters. The *data panel* option can be exercised

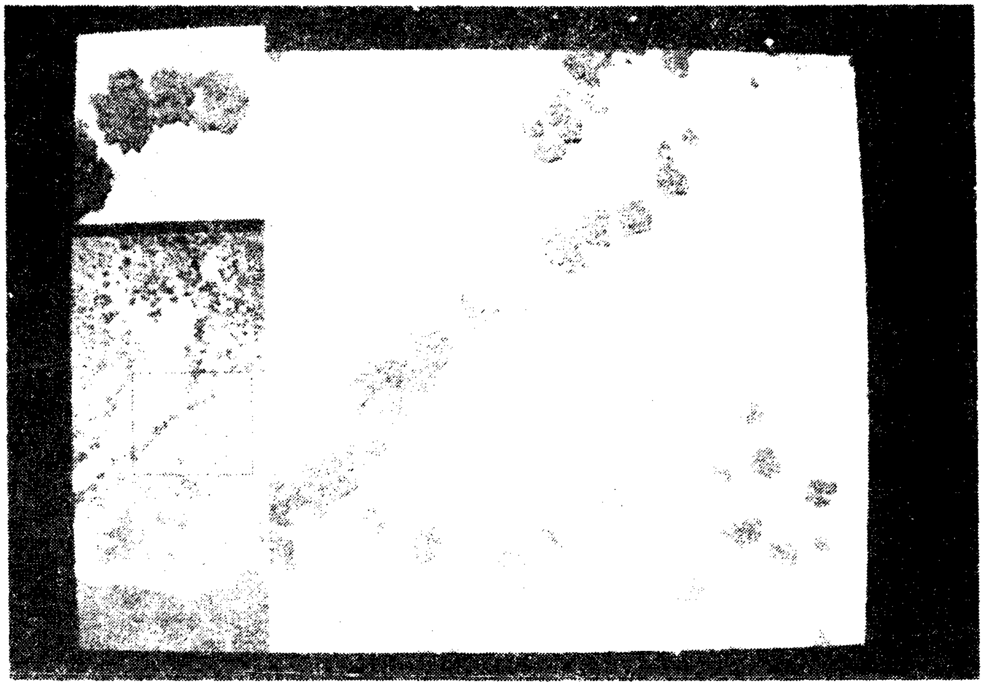


Figure 4.17 Two-Dimensional Aerial Photo Display

Figure 4.18 Two-Dimensional Aerial Photo Display

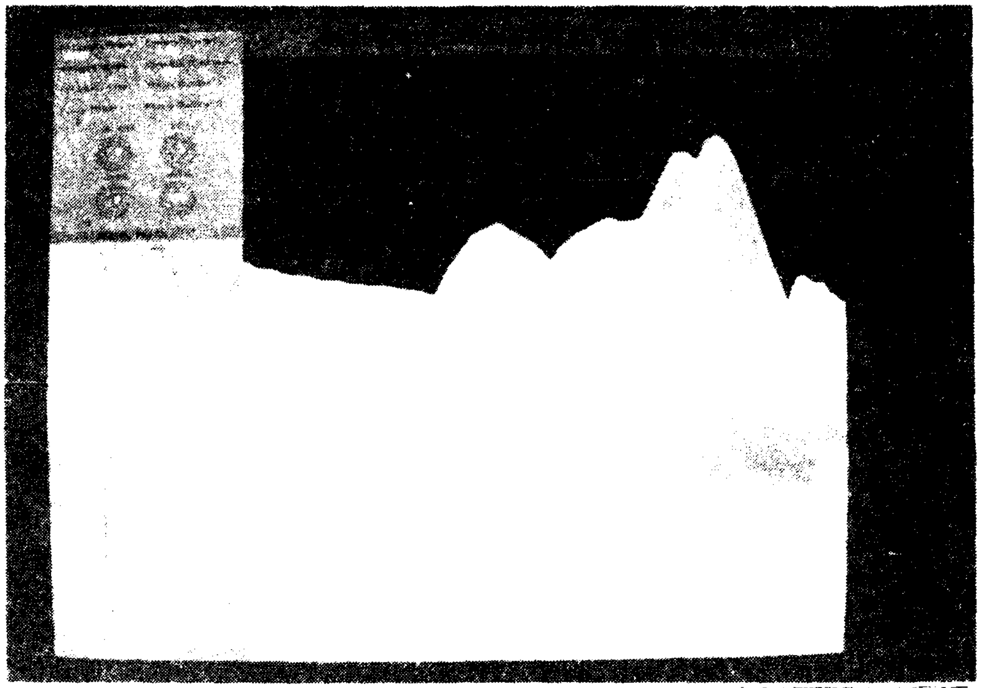


Figure 4.19 Three-Dimensional Perspective View

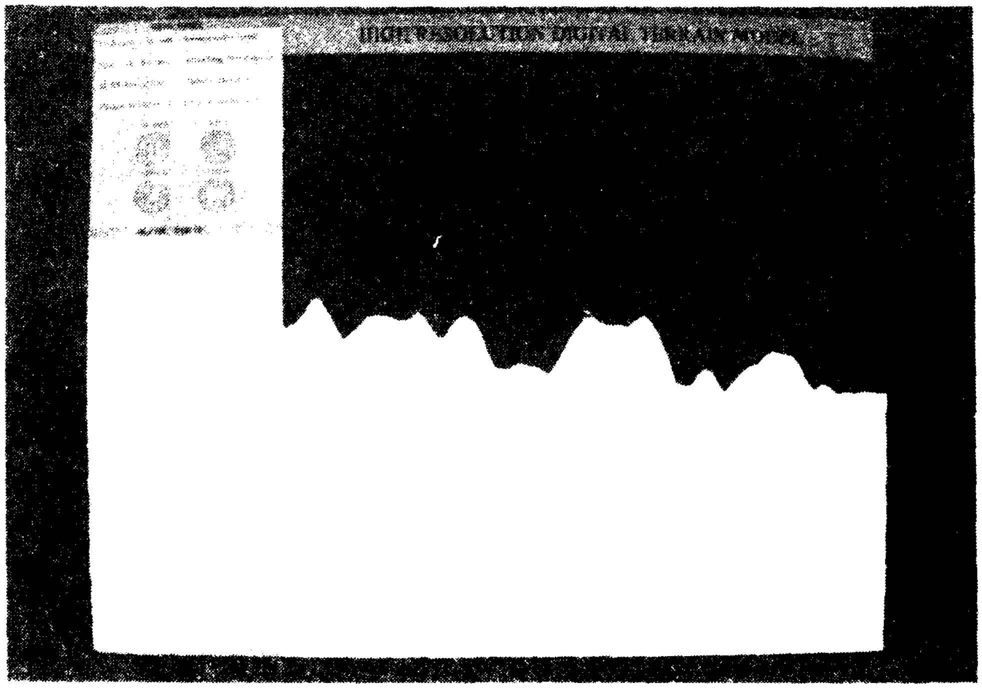


Figure 4.20 Three-Dimensional Perspective View

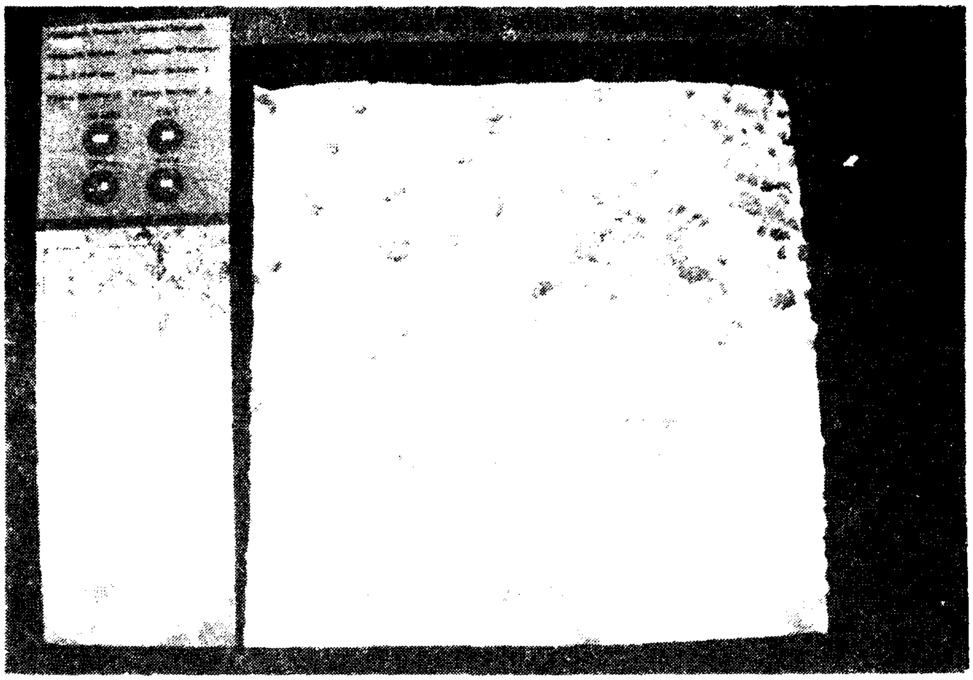


Figure 4.21 Data Panel

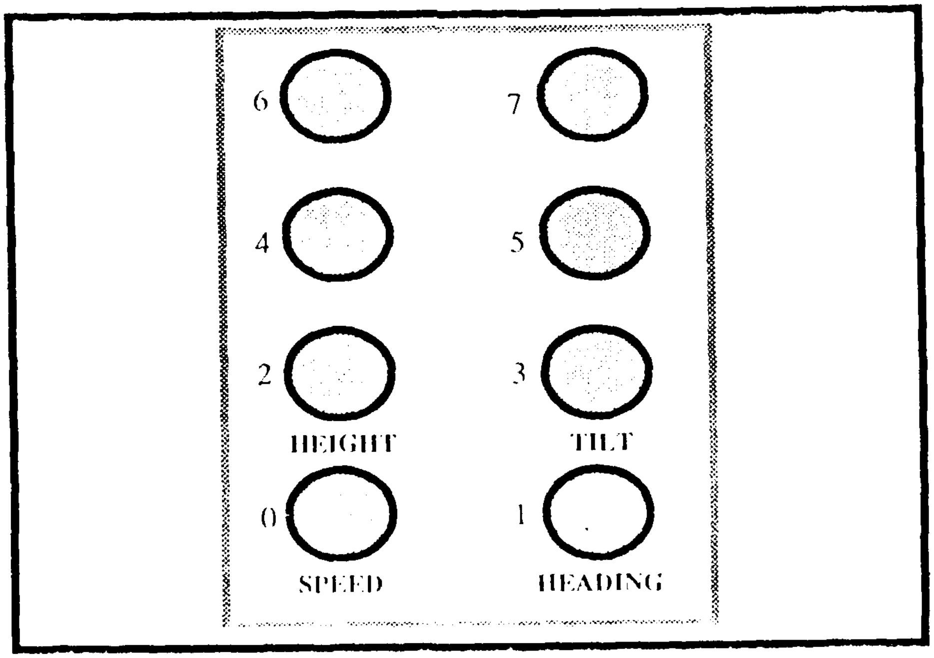


Figure 4.22 IRIS Dial Box

any time during simulator execution. The only time it displays any useful information, however, is when the simulator is in the three-dimensional terrain drawing mode.

5. Magnifier

The main menu's *magnifier* option provides a magnification or zoom capability. Selecting the *magnifier* option directs magnified object output to the magnify window in the upper-left corner of the screen. Figures 4.23 and 4.24 depict examples of the system's magnification capability. Any object on the screen can be magnified. By using the mouse to place the cursor over an object, the user can view the magnified object in the magnify window. The magnification factor is displayed in the window's title. This factor can be increased, decreased, or reset to its initial value by selecting the appropriate roll-off-the-side menu option to the main menu's *magnifier* option. Note that the magnification factor is initialized to two. Attempting to decrease this factor has no effect. There is no upper limit to the magnification factor; however, a magnification factor greater than six provides little additional information. Also, turning the magnifier off "freezes" the last magnified object. The *magnifier* option can be exercised any time during simulator execution.

6. Gamma Ramp Adjustment

The main menu's *gamma ramp* option permits the user to vary the gamma correction factor of the system. Changing this factor effects the color display of the entire screen. This is an extremely useful technique for highlighting or accenting certain terrain features. In essence, it permits the identification of features that would not normally be seen. Figures 4.25 and 4.26 depict this capability. The gamma factor can be increased, decreased, or reset to its initial value by selecting the appropriate roll-off-the-side menu option to the main menu's *gamma ramp* option. Note that the gamma correction factor equals one when the system begins execution. Attempting to increase this factor more than ten times or decrease this factor below its initial value

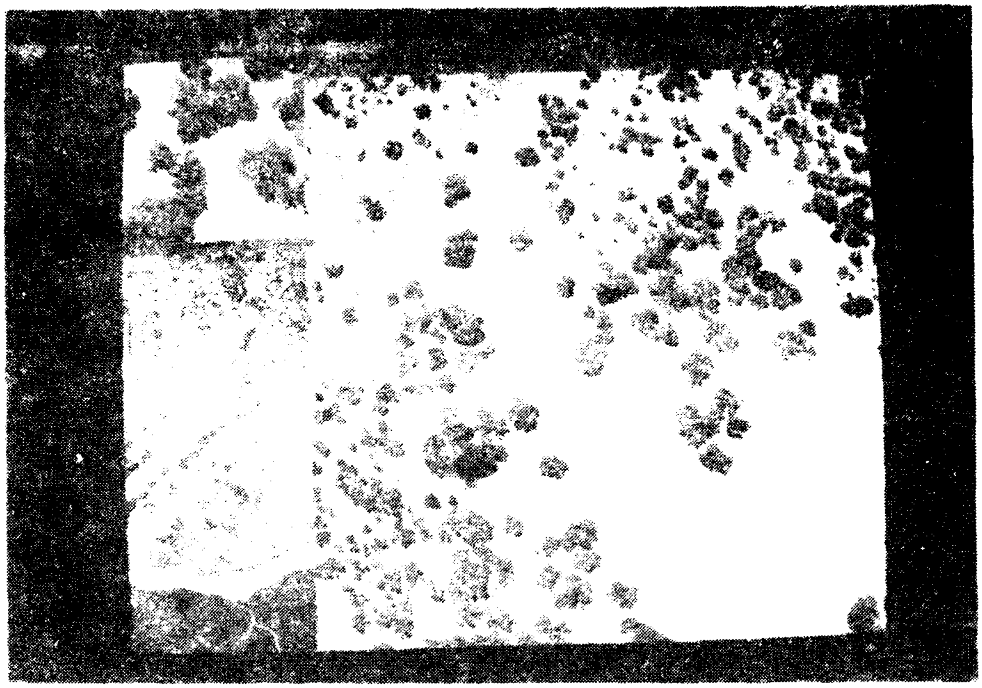


Figure 4.23 Magnification of Two-Dimensional Terrain

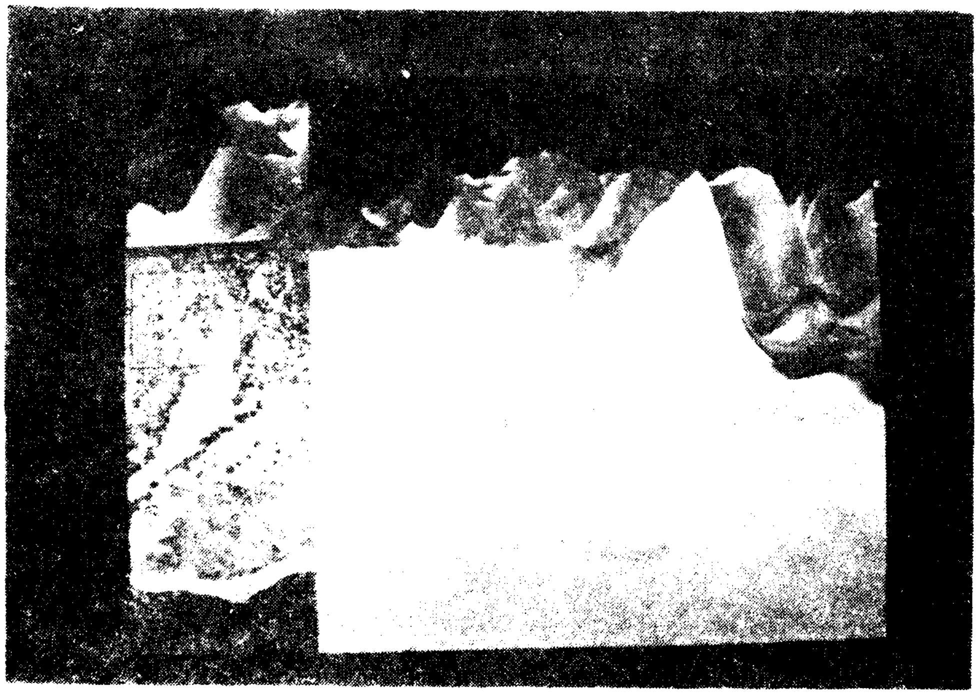


Figure 4.24 Magnification of Three-Dimensional Terrain

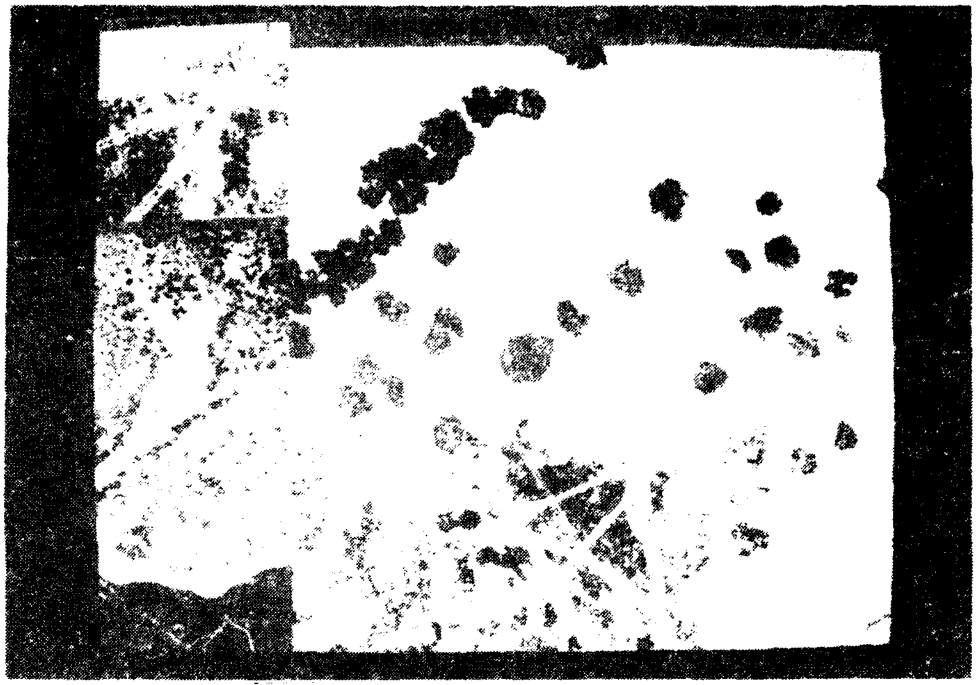


Figure 4.25   Gamma Modification of Two-Dimensional Terrain

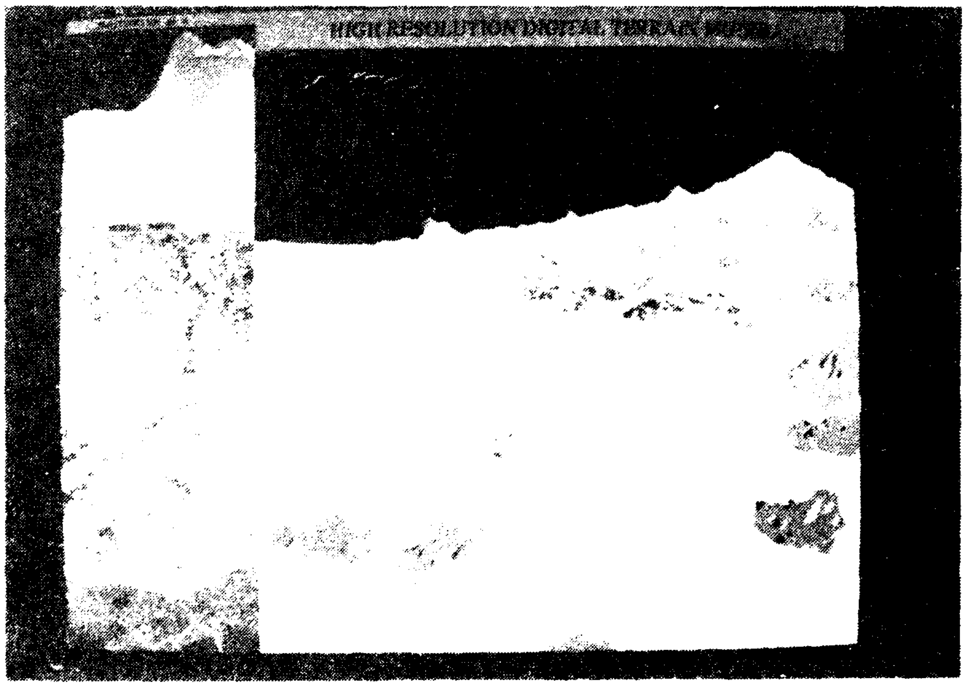


Figure 4.26   Gamma Modification of Three-Dimensional Terrain

produces no visible effects. The *gamma ramp* option can be exercised any time during simulator execution.

7. Screen Dumps to Laser Printer

The main menu's *camera* option provides the capability to print a selected portion of the screen on a laser printer. Selecting the *camera* option opens a window that is placed over the left corner of the area to be printed with the mouse. The window is then sized by selecting and holding the right mouse button while dragging the mouse. Releasing the mouse button after sizing the camera window takes a "picture" of the contents of that window which is sent, in Postscript format, to a file. The file is automatically transferred, via Ethernet, to our VAX computer which in turns executes a print command to a Postscript printer. The file transfer is required in order to access the laser printer that is linked to the VAX computer system. Figure 4.27 depicts a laser printed image of the screen. Note that the *camera* routine automatically scales a "picture" that is too large to print on the standard 8.5 x 11 inch paper used in laser printers. Once the *camera* option is selected, the user must snap a picture; there is no way to abort or exit from this option. The *camera* option can be exercised any time during simulator execution.

8. Legend

The main menu's *legend* option display's the elevation contour map's legend in the upper-left window. The legend depicts the colors representing 16 contour intervals that were calculated by dividing the difference of the maximum and minimum database elevations by 16.

9. Change Color

The *change color* menu option allows the elevation contour map and legend to toggle between two color ramps, a gray ramp and a brown ramp. Both ramps are created during program initialization by using the window manager's multimap mode.

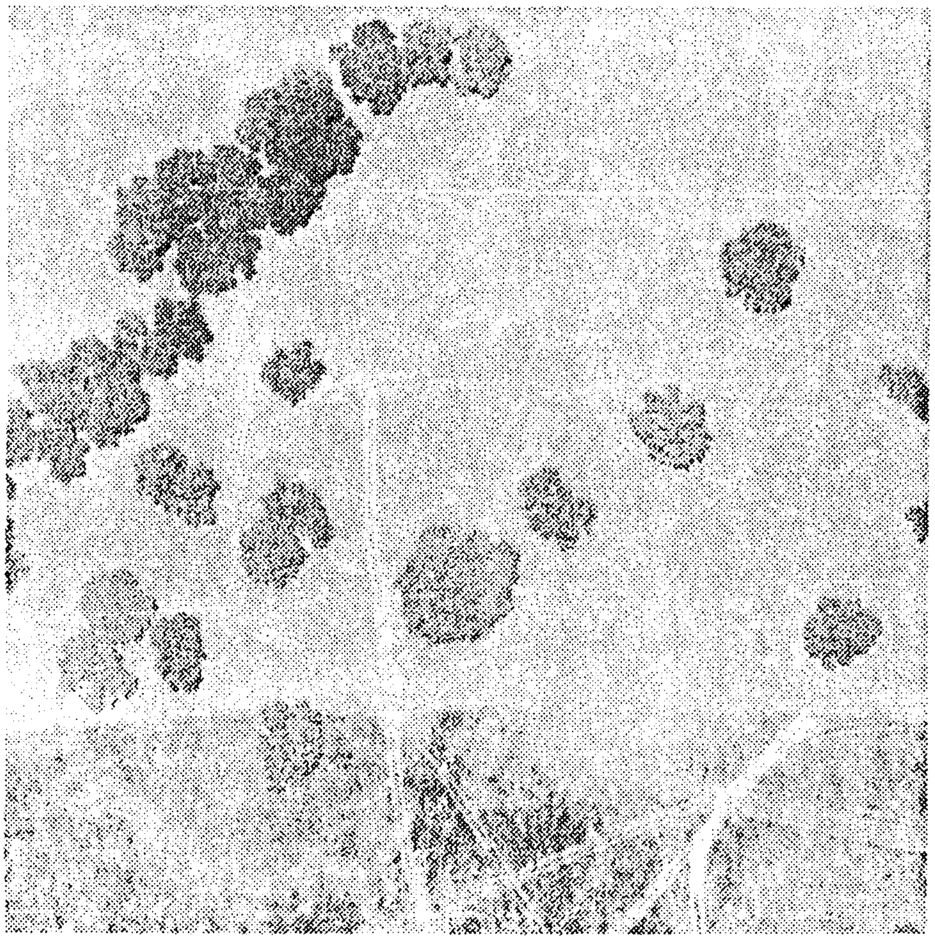


Figure 4.27 Laser Printed Image of Screen

The change color menu option simply changes the color map out from under both windows; switching from gray to brown and brown to gray. Changing the color map out from under the drawing avoids redrawing the entire scene with the new color ramp. Therefore, we incur no drawing time cost.

10. Swap Elevation/Photo

The double-buffered contour map window allows the black and white aerial photo of the entire database to be drawn in the back buffer. Since the elevation contour map and the aerial photo only need to be drawn once, we found that double buffering enables us to draw each in a separate buffer; the *swap elevation/photo* menu option permits toggling between the two buffers with no drawing time cost.

11. Termination

Program execution is terminated with the *exit* option in the main menu. Upon termination, all windows are closed, the system's color map is restored, and the system's gamma ramp is reset.

V. SYSTEM EVALUATION

A. PERFORMANCE MEASUREMENTS

1. Complexity/Speed

The HRDTM simulator operates at various levels of drawing complexity which provides an excellent means for evaluating the supporting IRIS graphics hardware. Drawing complexity, in this instance, refers to the number of polygons (triangles) drawn; the more polygons drawn, the more complex the view. This complexity varies with missile height and tilt angle. Essentially, the complexity increases as both the height and tilt increase because more and more terrain comes into view; therefore, more terrain must be drawn. Note that this is not always true because of the display algorithm that is used. This is noticeable in the performance measurements depicted in Tables 5.1, 5.2, 5.3, and 5.4. These measurements were taken over a wide range of viewing angles, but this does not appear to have effected system drawing performance in any manner. Our measurements indicate that system drawing performance is in the neighborhood of 36,000 to 41,000 Gouraud shaded triangles per second.

2. Realism

Judging the realism of the images generated by the system is a very subjective process. Our evaluation does account for the fact that we have personally surveyed the area of Fort Hunter-Liggett, California that comprises our database. Having visited and photographed the area, we can match generated terrain images with black and white photos. We would have had considerable difficulty doing this if we would not have surveyed the area prior to our system evaluation. This, however, only applies to photos taken at ground level. Figures 5.1 and 5.3 depict 35 mm black and

TABLE 5.1 HRDTM PERFORMANCE MEASUREMENTS ON IRIS 4D/70GT

	TILT ANGLES (DEGREES)			
	1	20	45	89
HEIGHT (METERS)	1	1	1	1
SECTORS (METERS)	12	12	24	72
DRAW DISTANCE (METERS)	685	685	72	2
FRAMES PER SECOND	0.93 - 1.14	0.89 - 1.14	4.5 - 5.7	20 - 32
TRIANGLES PER FRAME	33 - 40,000	33.5 - 40,400	7 - 8,000	1,300 - 1,600
TRIANGLES PER SECOND (AVG)	37,410	37,073	37,950	36,800

TABLE 5.2 HRDTM PERFORMANCE MEASUREMENTS ON IRIS 4D/70GT

	TILT ANGLES (DEGREES)			
	1	20	45	89
HEIGHT (METERS)	100	100	100	100
SECTORS (METERS)	12	12	24	72
DRAW DISTANCE (METERS)	685	685	685	217
FRAMES PER SECOND	1.35 - 1.68	1.38 - 1.65	0.68 - 0.78	0.90 - 0.97
TRIANGLES PER FRAME	22.5 - 27,300	22.6 - 26,800	49.5 - 53,300	41.8 - 44,200
TRIANGLES PER SECOND (AVG)	36,990	37,137	37,427	40,546

TABLE 5.3 HRDTM PERFORMANCE MEASUREMENTS ON IRIS 4D/70GT

	TILT ANGLES (DEGREES)			
	1	20	45	89
HEIGHT (METERS)	200	200	200	200
SECTORS (METERS)	12	12	24	72
DRAW (METERS) DISTANCE	685	685	685	436
FRAMES PER SECOND	1.33 - 1.67	1.32 - 1.63	0.68 - 0.76	0.32
TRIANGLES PER FRAME	22.8 - 27,800	22.7 - 26,450	48.5 - 53,600	128.5-129,300
TRIANGLES PER SECOND (AVG)	37,525	35,958	36,654	41,248

TABLE 5.4 HRDTM PERFORMANCE MEASUREMENTS ON IRIS 4D/70GT

	TILT ANGLES (DEGREES)			
	1	20	45	89
HEIGHT (METERS)	400	400	400	400
SECTORS (METERS)	12	12	24	72
DRAW (METERS) DISTANCE	685	685	685	685
FRAMES PER SECOND	1.36 - 1.67	1.36 - 1.60	0.7 - 0.79	0.25
TRIANGLES PER FRAME	22.4 - 26,500	23 - 27,500	47.6 - 52,700	148.3-149,800
TRIANGLES PER SECOND (AVG)	36,724	37,100	37,247	37,263

white photos of selected areas of Fort Hunter-Liggett, while Figures 5.2 and 5.4 depict the computer generated images of the respective areas.

We also evaluated the quality of images generated when the missile's viewing position increased in height and tilt angle. We discovered that gray-scale shading and coloring offers extremely little vegetation and cultural feature information from a vantage point below 100 meters. As depicted in Figures 5.2 and 5.4, trees, shrubs, rocks, and other features all appear as shaded hills; essentially, blending with the terrain. At 100 meters, as depicted in Figure 5.5, trees begin to come into focus and roads can be detected at a tilt angle of 25 degrees. Vegetation and other terrain data becomes more easily discernable, as shown in Figure 5.6, when the tilt angle exceeds 55 degrees. The problem with exceeding 55 degrees tilt, however, is that the three-dimensional drawing effect is lost. At tilt angles this great, the display appears as a two-dimensional black and white photo.

The quality of the two-dimensional black and white display is outstanding. Figures 5.7 and 5.8 depict examples. We discovered the resolution so good that we could display every other point in the database and still not lose any discernible amount of information. The fast pixel access and display functions provided in the system's graphics library appear to display the images almost instantaneously.

B. SYSTEM LIMITATIONS

1. Simulator Limitations

The HRDTM simulator was designed and implemented quickly in order to investigate the possibility of generating realistic images of terrain from processed aerial photo databases. It served this purpose well, but the rapid prototype processing imposed some restrictions and limitations on the system.

The user is restricted to flying his FOGM in a selected operating area of 288 meters x 288 meters. Ideally, the user should be able to select a start point for his missile and then he should be able to fly throughout the entire database.

Figure 5.1 Fort Hunter-Liggett Terrain Photograph

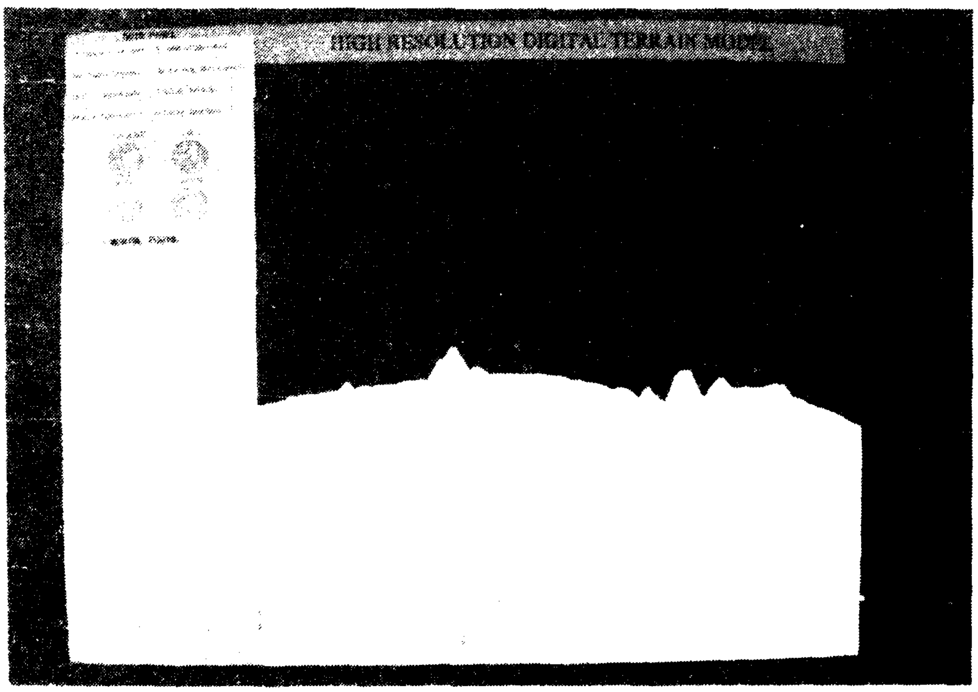


Figure 5.2 Computer Generated Perspective View

Figure 5.3 Fort Hunter-Liggett Terrain Photograph

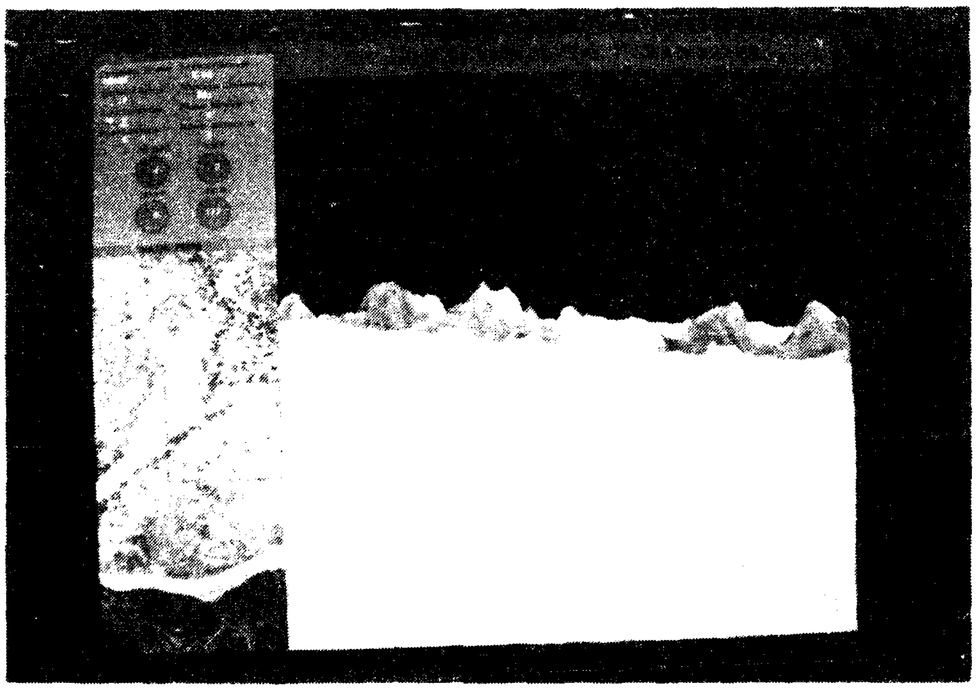


Figure 5.4 Computer Generated Perspective View

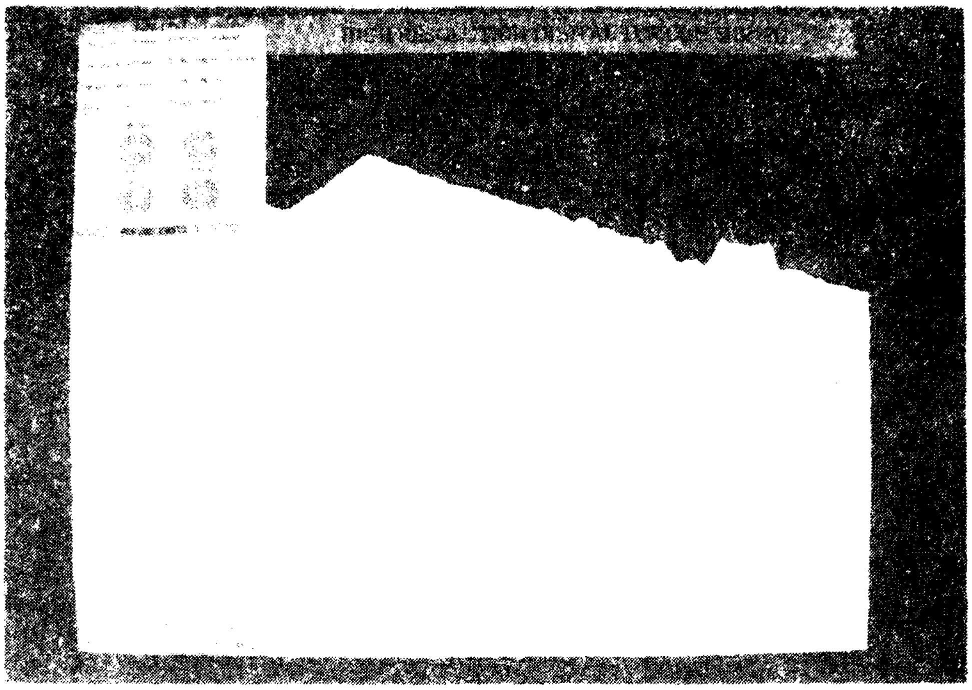


Figure 5.5 Three-Dimensional View with 25 Degree Tilt

Figure 5.6 Three-Dimensional View with 55 Degree Tilt

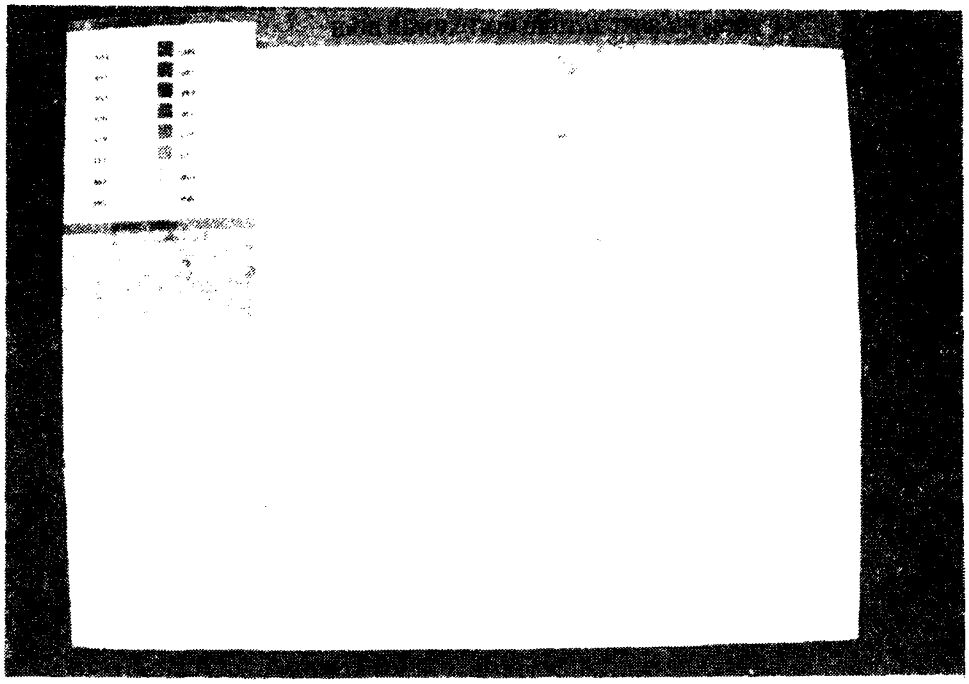


Figure 5.7   Computer Generated Two-Dimensional View

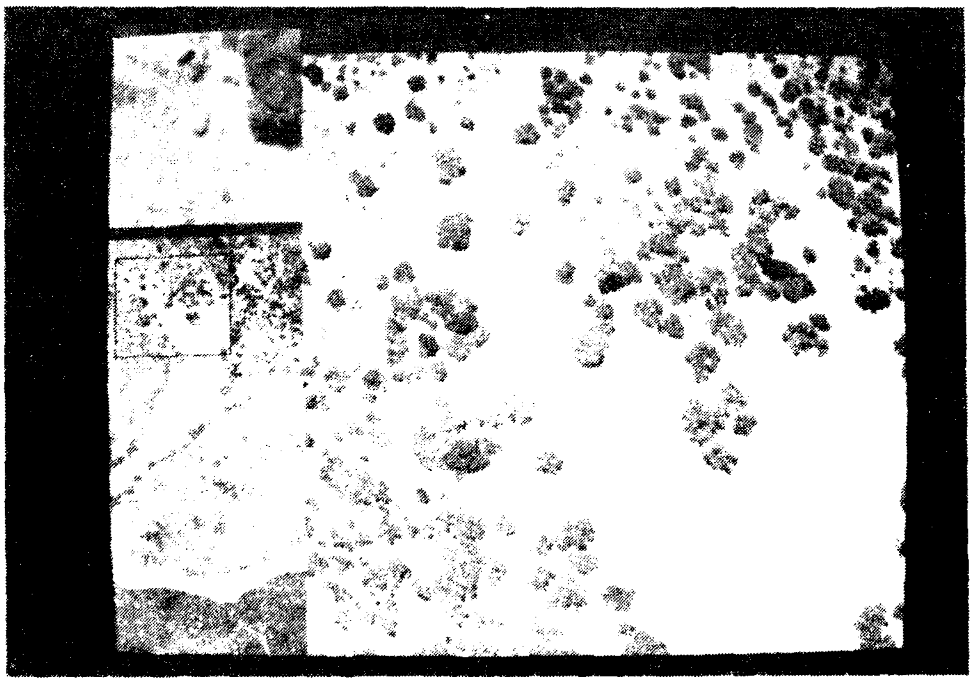


Figure 5.8   Computer Generated Two-Dimensional View

The system also lacks the ability to output the missile's location in UTM grid coordinates. The layout of the database does not contribute to the development of a simple position-location algorithm; therefore, it was not pursued.

2. Database Anomalies

The first anomaly to appear in the terrain display during a system run is caused from terrain elevation database inaccuracies. Very hilly terrain is displayed in areas that are located in extremely light portions of the aerial photo. Roads, for instance, appear as though they are covered with boulders. Figure 5.9 depicts such an example. In our estimate, this is caused from the database processing program's inability to highly correlate extremely light areas on stereo pairs.

The shadows that are present in our overhead photos create a second anomaly. When viewing the three-dimensional perspective views from certain angles, it appears as though the gray-scale data was "shifted" during the coloring process. In other words, sides of trees appear very lightly colored instead of dark. This is not the case, however. This shift appears because one side of the tree, in this case, was brightly lighted from the sun, while terrain on the other side is dark from the tree's shadow.

The other anomaly that appears in the terrain display is also caused from inaccuracies in the terrain elevation database. A "wall of terrain" occurs along two edges of the database. Figure 5.10 depicts an example of this problem. Once again, we attribute this problem to the processing techniques used to create the database. We believe that this was caused by an averaging methodology that was used on a group of pixels (or points) while scanning an aerial photo. When the scan reached the edge of the photo, the processing algorithm appears to have averaged the remainder of its scan lines with the last piece of data that it obtained. As a result, the outer two edges of the database consist of duplicated data. This duplicate elevation data appears as a wall when displayed.

Figure 5.9 Database Anomalies

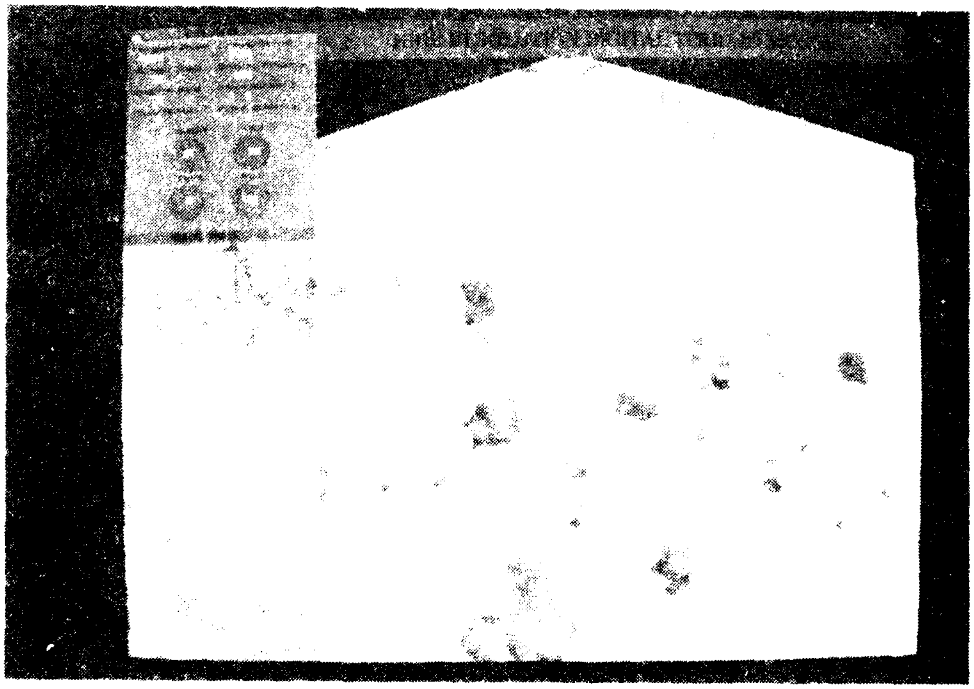


Figure 5.10 Database Anomalies

VI. SUMMARY

A. CONCLUSIONS

This study originated from our desire to generate more realistic images than those currently generated by three-dimensional visual simulators such as the Moving Platform Simulator [Ref. 2]. We discovered that realism was greatly enhanced by increasing the resolution of the terrain elevation database. As we expected, however, increasing resolution slows the simulator drastically. One point that we found interesting was that 0.3 meter resolution provided us no more noticeable information than 0.6 meter resolution. Our database storage requirements can be cut in half if we use a 0.6 meter resolution database, and we will not sacrifice much information loss. Another important result of our study is that we discovered that texturing the terrain with corresponding aerial photo reflectance data provides almost no information until we view the terrain from almost directly overhead. In the overhead case, however, we lose the height aspect of the three-dimensional drawing; therefore, we are better off displaying the two-dimensional aerial photo, a much faster process.

In the area of graphics workstation performance evaluation, we can conclude that the IRIS 4D/70GT draws, on the average, 37,000 Gouraud shaded triangles per second. The HRDTM simulator generates, in its worst case, an overhead view of a 288 meter x 288 meter segment of terrain requiring approximately 150,000 Gouraud shaded triangles. Therefore, the HRDTM simulator is capable of only a 0.25 frames per second drawing rate during a worst-case scenario. Thus, the current drawing rate is insufficient to drive the HRDTM simulator at our real-time requirement of two to three frames per second.

B. FUTURE RESEARCH

The high-resolution database does, however, offer an excellent source for other research. One such use would involve integrating only the terrain elevation portion of the HRDTM database directly into MPS. Since standard Defense Mapping Agency (DMA) databases normally provide only 100 meter resolution and the special Fort Hunter-Liggett database provides only 12.5 meter resolution, the high-resolution terrain elevation data file may provide an excellent alternative database for improving Moving Platform Simulator accuracy.

Since coloring and shading the terrain with its corresponding aerial photo gray-scale data provided little cultural feature and vegetation information, there is still a requirement to display this information. Therefore other research could involve using both the elevation and reflectance data files along with artificial intelligence techniques to determine cultural features. Once these features are identified, one may then generate synthetic cultural features in the display.

Another area of possible research is the management of large terrain databases. Terrain database design, file management, and storage will play a crucial role in future systems that will have to access and display huge amounts of information. Optimizing database design for ease of file access and storage offers an excellent opportunity for research. CDEC has recently obtained an 80 square kilometer high-resolution terrain elevation and gray-scale database of the entire Fort Hunter-Liggett area. This database can provide a starting point for such database research.

Real-time generation of realistic two and three-dimensional terrain displays remains an exciting area of research. Research is limited only by the resolution of digital terrain elevation databases, central processor memory, and faster computer graphics hardware. Even now, however, great strides are being made to overcome each of these limitations. Database resolution is better and more available than ever before. Faster and more capable graphics hardware is also becoming more available and affordable. As each of these restrictions is overcome, greater and greater

achievements will be realized. In light of these advances and continuing research, we feel that the work completed in this thesis will provide an excellent stepping stone for future research.

LIST OF REFERENCES

1. McGhee, R. B., Ross, R. S., Smith, D. B., Streyle, D. G., and Zyda, M. J., "Flight Simulators for Under $100,000," *IEEE Computer Graphics & Applications*, pp. 19-27, January 1988.

2. Naval Postgraduate School, Technical Report NPS52-89-004, *Meaningful Real-Time Graphics Workstation Performance Measurements,* Fichten, Mark A., Jennings, David H., and Zyda, Michael J., November 1988.

3. Smith, Douglas B., and Streyle, Dale G., *An Inexpensive Real-Time Interactive Three-Dimensional Flight Simulation System*, M.S. Thesis, Naval Postgraduate School, Monterey, California, June 1987.

4. Oliver, Michael R., and Stahl, David J., *Interactive, Networked, Moving Platform Simulators,* M.S. Thesis, Naval Postgraduate School, Monterey, California, December 1987.

5. Silicon Graphics Inc., *IRIS User's Guide, MEX Window Manager*, Mountain View, California, 1987.

6. Silicon Graphics Inc., *4Sight User's Guide*, v. 1, Mountain View, California, 1988.

7. Silicon Graphics Inc., *IRIS User's Guide, GT Graphics Library User's Guide*, Mountain View, California, 1987.

8. Telephone conversation between Dr. Ignace Liu, Combat Developments Experimentation Center (CDEC), Fort Ord California, 93941 and Captain James J. Zanoli, 2 May 1989.

9. *Background and Current Status, 1988*, three page paper highlighting the background of GeoSpectra Corporation, prepared by GeoSpectra Corporation, P.O. Box 1387, 333 Parkland Plaza, Ann Arbor, Michigan 48106.

10. *ATOM® Version 3.0*, one page paper describing the Automatic Topographic Mapper, prepared by GeoSpectra Corporation, P.O. Box 1387, 333 Parkland Plaza, Ann Arbor, Michigan 48106.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Dr. Michael J. Zyda Naval Postgraduate School Code 52, Department of Computer Science Monterey, CA 93943-5100	200
4.	Cpt. William O. Breden 30829 Lincoln Drive Granger, IN 46530	2
5.	Cpt. James J. Zanoli 11 Meadowvale Drive Cheswick, PA 15024	2
6.	Superintendent Naval Postgraduate School Code 012, Research Administration Monterey, CA 93943-5000	1
7.	Commandant of the Marine Corps Code TE 06 Headquarters United States Marine Corps Washington, D.C. 20380-0001	1
8.	Mr. Mike Tedeschi United States Army Combat Developments Experimentation Center Attention: ATEC-D Fort Ord, CA 93941	5